Marksman GRPS
(Golden River Protocol System)

User Manual

GR560243 Issue 2.2
February 1998

Release Notes (for more details, see Appendix 5)

Issue 2.0	Extracted from *Marksman 660* User Manual Issue 1 and completely rewritten. This issue corresponds to *Marksman 660* firmware version 1.90.
Issue 2.1	Corresponds to *Marksman 660* firmware version 1.95. New commands added: CLOCK, TEST SYSAUDIT and TEST LPSTATUS. Other minor language and firmware updates. Minor editorial changes.
Issue 2.2	Includes all GRPS commands, including *Marksman 410*. Corresponds to *Marksman* firmware version 1.98. New commands added in Chassis Height (CH), Tyre Contact Width (TCW) and Weight (WT) series. Full Year 2000 compliance. Other minor language and firmware updates. Minor editorial changes.

Introduction

This is the User Manual for **Marksman GRPS**, the **Golden River Protocol System** as implemented in the Golden River *Marksman* series of traffic counters and classifiers.

If you cannot find the information you need in this Manual, please contact us for assistance, asking for the Technical Support Department.

Golden River Traffic Ltd
Churchill Road
Bicester
Oxfordshire OX6 7XT
UK

	UK	**Outside the UK:**
Phone-Main	01869 362800	+44 (0) 1869 362 800
Phone-Support	01869 362802	+44 (0) 1869 362 802
Fax	01869 246858	+44 (0) 1869 246 858
		Or contact local Golden River representative.
E-mail:	support@goldenriver.com	
Internet:	http://www.goldenriver.com	

© 1998 GOLDEN RIVER TRAFFIC LIMITED

ALL RIGHTS RESERVED

Copyright in the whole and every part of this Manual belongs to Golden River Traffic Limited. This Manual may not be used, sold, transferred, COPIED or REPRODUCED in whole or in any part in any manner or form or in or on any media to any person other than with the prior written permission of Golden River Traffic Limited.

Golden River Traffic Limited will supply further copies of this Manual to an end-user on request. A nominal charge will be made to cover printing and distribution costs.

Contents

Introduction i

How to Use this Manual v

Availability of GRPS Commands vi

WHAT IS GRPS? 1

What GRPS Offers 1

FINDING THE COMMAND YOU WANT 3

Availability of GRPS Commands 3
'Technical' Commands 3

Information Flow 4

Vehicle Detection Commands 6

Detector control 6
Time-tagged events 6
Tyre Contact Width (Marksman 410 only) 6
Weigh-in-Motion (WIM) 6
Other settings 7
Environmental 8

Vehicle Processing Commands 8

Vehicle Processor control 8
Vehicle Processor Monitoring 8

Data Storage Commands 9

Data recording 9
Recording control 9
Date, time and location 9
File handling 10

Input / Output Control Commands 10

Data terminal 10
File transfer 10
Output ASCII format 10
Test Commands 11

Remote Interface Commands 11

Other Commands 11

OPERATOR INTERFACE 13

Conventions in this Manual 13
Syntax descriptions 13
Valid ranges and error messages 14
Default values 15
CAUTION notes 15

Password Protection for Remote Access (*Marksman 660*) 15

Input Prompts 16

Entering Commands 16
Typing mistakes 17
Error messages 17
Repeating the previous command 17
Filenames 17

Keypad interface 18
Availability of Keypad Commands 18

GRPS COMMANDS: ALPHABETICAL REFERENCE 25

Appendix 1 – GRPS Error and Warning Codes 285

Appendix 2 – GRPS ASCII Print Formats 290

Appendix 3 – ASCII Codes 302

Appendix 4 – Data Module Commands 303

Appendix 5 – Release Notes since Issue 2.0 307

How to Use this Manual

This manual is divided into four major parts:

- **About GRPS**

  The origins and major features of GRPS.

- **Finding the Command You Want**

  GRPS commands classified into major functional areas.

- **Operator Interface**

  Using GRPS with an external PC terminal or a built-in keypad and LCD display.

- **GRPS Commands: Alphabetical Reference**

  All the GRPS commands explained in alphabetical order.

Appendices 1–4 cover Error Messages, examples of output formats, ASCII codes and the special commands for the Golden River *Data Module*.

You should read this **GRPS Manual** in conjunction with the following manuals from Golden River:

- ***User Manuals*** for your traffic survey instruments
- ***Counting and Classification Highway Manual***
- ***Weigh-in-Motion Highway Manual.***

Availability of GRPS Commands

This manual includes all of the GRPS commands, but those available on your particular model of the *Marksman* will only be part of the full set.

If you attempt to use a command that is not available on your model of the *Marksman*, you will see an error message, most probably:

```
Error 99    Command unavailable on this machine
```

This means that the command you have just attempted is a valid GRPS command, but is not available on your *Marksman*.

- **Marksman 660**

  The *Marksman 660* has all the GRPS commands except the specialised TCW– series (*Marksman 410* only). However, your particular *Marksman 660* will only respond to commands that apply to the sensor and switch cards that have been fitted. For example, a *Marksman 660* that has no Weigh-in-Motion sensors will not respond to the WT– series of commands.

- **Marksman 360**

  The *Marksman 360* is a simplified version of the *Marksman 660* which has only loop sensors and can only count vehicles. Therefore the *Marksman 360* will not respond to GRPS commands for other sensors or for vehicle-by-vehicle monitoring.

  Also the *Marksman 360* has only one **Com** port and does not have certain other facilities, notably password security.

- **Marksman 400 and Marksman 410**

  The *Marksman 400* and *410* only have tube sensors, and therefore will not respond to GRPS commands for other sensors.

  Also the *Marksman 400* and *410* have only one **Com** port and do not have certain other facilities, notably password security.

- **Marksman 410**

  The *Marksman 410* is the only model that has Tyre Contact Width tube sensors, so only the *Marksman 410* will respond to the TCW– series of GRPS commands.

  Certain other GRPS commands have added functions when used with the *Marksman 410*.

What is GRPS?

The **Golden River Protocol System (GRPS)** is a standardised language for communicating with traffic counters, classifiers, pollution monitors and weigh-in-motion classifiers.

The Golden River *Marksman* series were the world's first microcomputer-controlled traffic counters. Since their introduction in 1978 there has been a proliferation of traffic counting equipment, but users have been inconvenienced by the lack of standardisation in keyboard interfaces, serial port connections and command structures.

What GRPS Offers

In response to this need, Golden River Traffic has developed a standardised user interface and protocol to be implemented in its equipment.

GRPS offers:

- A single command set and protocol which can apply to any traffic monitoring device
- A consistent, easily understood user interface
- Generic standard output formats
- Standard parameters for RS-232 communications with external devices such as modems and terminals
- Embodiment of good practice in sensor technology and layouts.

Golden River hopes that readers will write or call with comments and suggestions, to further the development of GRPS as an effective industry standard

Finding the Command You Want

The next few pages will help you to find the GRPS command you want to use, from one of the following categories:

- Vehicle detection
- Vehicle processing
- Data storage
- Input/output control
- Remote interface security
- Test commands
- Other functions.

You can then find the relevant command in the main alphabetical list. You may also need to read the general introduction to GRPS operator interface and command structure beginning on page 13.

Availability of GRPS Commands

This manual includes all of the GRPS commands, but those available on your particular model of the *Marksman* will only be part of the full set.

If you attempt to use a command that is not available on your model of the *Marksman*, you will see an error message, most probably:

```
Error 99    Command unavailable on this machine
```

This means that the command you have just attempted is a valid GRPS command, but is not available on your *Marksman*.

'Technical' Commands

This symbol indicates a 'technical' command which very few users will ever need. Incorrect use of 'technical' commands may result in inaccurate recording or loss of data, so you are advised to consult Golden River or your national Golden River representative before using them.

The same symbol appears on the detailed descriptions of commands classified as ‘technical’ commands, in the Alphabetical Command section beginning on page 25.

Information Flow

Figure 1 overleaf shows the information flows through the machine and into data storage. The different regions of Figure 1 form the basis for classifying GRPS commands.

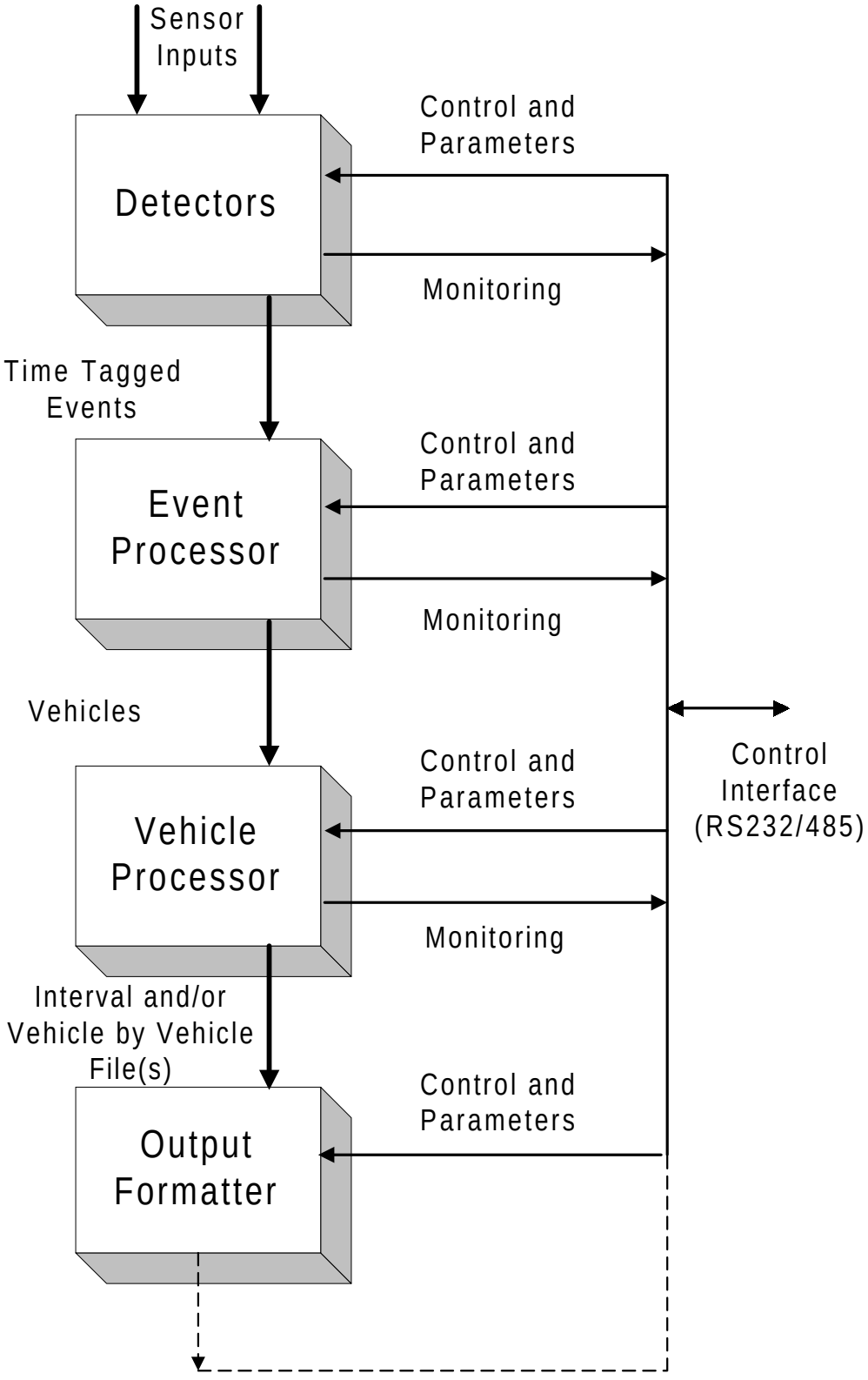
The hardware sensors produce time-tagged events (TTEs). The TTEs are passed to the software Event Processor which identifies individual vehicles and passes this information along with the TTEs to the Vehicle Processor software.

The Vehicle Processor carries out calculations on the TTEs to determine such parameters as speed, length, number of axles and vehicle classification. The output is routed to either an Interval (INT) data file or a Vehicle-by-Vehicle (VBV) data file. If VBV recording has been selected, the associated TTEs are logged in parallel with the vehicle-by-vehicle data.

Not all of the data collected by the Vehicle Processor need to be recorded. A range of ‘filter’ commands allow Interval, VBV and TTE data to be recorded selectively.

Other options include On-Site Printout of results as they are generated, and actuation of switch contacts to operate external equipment. Both of these can be carried out selectively using ‘filter’ commands.

Figure 1: Information flow and categories of commands



Vehicle Detection Commands

Commands for the sensor-detector system which produces unprocessed time-tagged events

Shaded commands are 'technical' – see page 3.

Detector control

DETOFF	Inhibit all detectors
DETON	Enable and reset all detectors
MONITOR	Monitor detector activity
CHANNELS	Lane to channel assignment
SWINPUT	For use with external sensors

Time-tagged events

OSP+	List TTE sensor readings
TTEFILTER	Selective TTE logging
SKEW	TTE timing

Tyre Contact Width (Marksman 410 only)

TCWCALIB	TCW sensor calibration
TCWNOISE	TCW sensor noise threshold
TCWTHRES	Cycle/motor vehicle threshold

Weigh-in-Motion (WIM)

WTAUTCAL
WTBINS
WTCAL
WTCALIB,
WTCALIB+
WTCALVEH
WTCYCLES
WTEVLEN
WTFILTER
WTHOLD
WTMAXAREA
WTMINAREA

WTMODE WTOVERS WTTHRES WTUNDERS WTWINDOW	

Other settings

AXFILTER AXSEPS	Axle detection by tube
EOVDSPEED	Crossing/overtaking vehicle detection
EOVDACC EOVDGAP EOVDSEP	End-of-vehicle detection, by acceleration, gap or axle separation
LPCALC LPCYCLES / LPSENS LPHOLD LPMODE LPOVERS LPSCAN LPTHRES LPUNDERS	Technical commands for loop detectors
CHAUTAVLOW CHAUTCNT CHCALC CHFIELD CHHIGH% CHUSRAVLOW	For ‘tuning’ a two-loop vehicle classification layout
CLS2CNT LPCLS2	For ‘tuning’ a two-loop EUR6 classification layout
PZFILTER PZTHRES	Piezo detection

Environmental

GAS	CO detection
GASCAL	
TEMP	Temperature measurement
TEMPCAL	

Vehicle Processing Commands

Commands which apply to the Vehicle Processor, which converts time-tagged-events into vehicle-by-vehicle (VBV) data

Vehicle Processor control

UNITS	Units of measure for input parameters
SENSORS	Define sensor layout in each lane
AXSEPS	Axle sensor separations
LPLENS	Length of loops in each lane
LPSEPS	Separation of loops in each lane
CLASS	Vehicle classification scheme
AXFACTOR	Average axles per vehicle ratio for tube/piezo counters
SWOUTPUT	Selective operation of switch contacts

Vehicle Processor Monitoring

OSP, OSP+	On-Site Printout of VBV and TTE data
OSPFILTER	Criteria for On-Site Printout
OSPTIME	OSP timeout
OSPVEH	

Data Storage Commands

Data recording

CHANNELS	Lane to channel assignment
INTSPEC	Interval recording criteria
INTERVAL	Recording interval
PEAKINT	Peak-period recording interval
ACTBINS	Axle count bins
LENBINS	Length bins
SPDBINS	Speed bins
TIMEBINS	Time bins (for gap and headway)
INTFILTER	Selective interval recording
VBVFILTER	Selective vehicle-by-vehicle recording
TTEFILTER	Selective time-tagged-event recording

Recording control

INTONOFF	Interval recording on/off periods
PEAKTIME	Daily peak-time periods
VBVONOFF	VBV recording on/off periods
STARTREC	Start recording immediately
STOPREC	Stop recording immediately
BREAK	Periodic closing of recording files
SUMMER	Summer-time start date
WINTER	Winter-time start date

Date, time and location

TIME	Time of day
DATE	Calendar date
DATEFORM	Date format
CLOCK	Time and date together
SITE	Site name
LOCATION	Site location description

GRIDREF	Site Grid Reference
HEADINGS	Site lane bearings

File handling

FILENAME	File name for recordings
MEMFULL	Action when memory full
DIR	List the data files in memory
CHMOD	Change the status of a file in memory
DELETE	Delete file in memory

Input / Output Control Commands

Commands which control communication with external devices, including data file transfer to and from the machine's memory

Data terminal

COM1 (COM2)	Serial port 1 (port 2) parameters

File transfer

PROTOCOL	Protocol for file transfer
UPLOAD	Upload file to machine in binary format
RETRIEVE / DOWNLOAD	Retrieve (download) file(s) in binary format
PRINT	Output files in ASCII format

Output ASCII format

PAGELEN	Screen scrolling
PRUNITS	VBV, OSP output units
EOLCHARS	End-of-line characters
EOFCHARS	End-of-file characters
EOPCHARS	Page length and end-of-page characters

Test Commands

TEST DISPLAY	Test LCD display

TEST FILE	Make test data files
TEST KEYBOARD	Test keypad operation
TEST LPSTATUS	Display loop status
TEST SYSAUDIT	Audit check on configuration
TEST VEHCOUNT	Test/reset vehicle counts
TEST VOLTS	Main battery voltmeter

Remote Interface Commands

Commands related to the modem serial port (COM2), including protection against unauthorised remote access.

COM2	Serial port 2 parameters
INTERFACE	Set level of access
PASSWORD	Passwords to change access level
TIMEOUT	Lock access automatically after use

Other Commands

ALL / HELP	Show all parameters and settings
LANGUAGE	English, French, German, Spanish
SERIAL	Read or change machine's serial number
SETUP	Automatic setup dialogue
STATUS	Show hardware status
CFGSAVE / CFGSTORE	Store the current configuration into a memory file
CFGLOAD	Load a configuration file from memory into machine
RESTART	Restart, with various levels of reset
REM	Comment or 'remarks' line
REPEAT	Repeat last command

Operator Interface

Conventions in this Manual

This manual uses various typefaces to denote commands, input and output. Generally it is assumed that you will be typing at the PC Terminal, but a separate section below describes the keypad interface.

When describing a command, its name is generally given in capitals, e.g. 'the INTSPEC command'.

In syntax descriptions, anything that you must type is given in bold lower case (except when capital letters are necessary), e.g. '**filename = file1234**' or '**sensors = LL**'. For further details, see below.

Default configurations are given in plain lower case, e.g. 'break = off'.

A response from the machine is given in a different typeface, e.g. `Checking directory, please wait ...`

Syntax descriptions

Many commands can be used in two different ways, either to change a machine setting or merely to display its present state. Thus there are often two headings in the command description, **Command Syntax** and **Display Syntax**.

Command Syntax uses a semi-formal description, for example:

break = hourly / daily / weekly / off / *Time*

'**break =**' is a required part of the command. Also required is one of the options separated by '/ ' symbols.

The options '**hourly**', '**daily**', '**weekly**' and '**off**' are shown in bold type, so you should type in one of them exactly as spelled above (without any quotation marks).

An additional option is shown above as *Time*. Words in italics like this have to be replaced by something that you must enter in a particular format. In this example, *Time* has to be replaced by a number of minutes, for example 720. The required format is always explained in the *Parameters* section immediately below the syntax statement.

Where words or options are enclosed in square brackets [ ], this means that everything between the brackets can be omitted.

Here is a more complex example:

actbins = *LowestBin* [*BinBoundary*...] *HighestBin*

This means that something **must** be substituted for the 'place-holders' *LowestBin* and *HighestBin*, but the square brackets around the intermediate [*BinBoundary*] mean that it is optional, or may not always be needed.

The dots ... following *BinBoundary* mean that more than one value can be inserted if required.

There is always at least one practical example to clarify the command syntax, with an even more detailed explanation if needed.

Display Syntax descriptions are much simpler, because the required command is often simply the command name (shown in bold lower case). There is always an *Example Response* from the machine in the different typeface, for example:

Display syntax	**actbins**
Example response	`ACTBINS = 0 1 2 3 4 5 6 99`

You can use the 'display' mode to verify that you have made a change correctly – simply type in the command name and the machine will respond with the current settings for that command.

Valid ranges and error messages

Where a command has one or more accompanying parameters, these usually have valid ranges, and attempted entries beyond these ranges will produce an error message.

Valid ranges are shown where applicable to a particular command.

Default values

Many commands have default parameter values, which are shown where appropriate. The RESTART command sets almost all parameters to their default values, the main exceptions being basic settings such as CLOCK.

Where no default is shown, this implies a blank, empty or null default value.

CAUTION notes

➲ **CAUTION**
Notes like this highlight common user errors.

Password Protection for Remote Access (*Marksman 660*)

In the *Marksman 660*, the INTERFACE command can be used to change the level of access at the remote serial port. The three available modes are:

- **Read-write**
- **Read-only**
- **Locked**.

Read-write mode is the normal mode of the machine, giving full access to all GRPS commands in either their **Command Syntax** and **Display Syntax** as documented in this User Manual.

In **Read-only** mode, many commands entered via the remote serial port are restricted to their **Display Syntax** only. The restricted commands are those that would change the machine settings or affect the stored data.

Read-write and **Read-only** modes can be password-protected using the PASSWORD command.

In **Locked** mode, the only functional command is INTERFACE itself – so that access can be changed to one of the two higher levels using the appropriate password.

The INTERFACE setting only affects access via the remote serial port. Local control via the PC Terminal or the keypad (if fitted) is always in full **Read-write** mode.

Input Prompts

When you connect a PC terminal to the serial port of a *Marksman*, the machine supplies an ‘input prompt’ and then waits for your input.

This prompt is a '>' character prefixed by a letter which indicates the current status of the machine. When the machine is first powered-up, the input prompt is `Q>`.

Here is the complete list of possible input prompts (in alphabetical order):

`B>`	Both Interval and VBV recording: in progress
`b>`	Both Interval and VBV recording: waiting until programmed start time
`D>`	Detectors operational
`I>`	Interval recording: in progress
`i>`	Interval recording: waiting until programmed start time
`Q>`	Quiescent, no detectors or recordings activated
`V>`	Vehicle-by-vehicle recording: in progress
`v>`	Vehicle-by-vehicle recording: waiting until programmed start time

Entering Commands

A command is contained on a single line of input, completed by pressing the **Enter** key. The first word in the line is always the command name, for example **dateform**.

Where an **=** sign is required, spaces before and after the **=** are optional. Any further parameters must be separated by one or more spaces; the number of spaces does not matter as long as there is at least one.

In this manual your typed input is always shown in lower case, but in general **GRPS commands are not case-sensitive** – you can input them in either lower-case or CAPITALS. Exceptions: remote access passwords and also parameters used with the command SENSORS **are** case-sensitive.

Typing mistakes

If you notice a mistake while typing a line, you can use the **Back-space** (←) key to erase characters back to the mistake, and then correct the mistake and re-type the rest of the line.

If you press the **Enter** key without noticing the mistake, you will see an error message instead.

Error messages

If your input was not correct, the machine will show an error message (Appendix 1) followed by a new input prompt. For example, if you mis-type the TIME command:

Input Q>**ti,e**

Response `Error 01 : Unrecognised Command`

If the machine recognises the command but not the rest of the line, you may see a variety of error messages such as:

`Error 03 : Invalid parameter (check case)`

Note that the error message only represents the machine's 'best guess' at the problem.

Repeating the previous command

To make the machine re-type your previous entry so that you can edit it, press **R** (for REPEAT) followed by **Enter**.

Filenames

GRPS uses MS-DOS filenames for compatibility with the PC Terminal, *Showman Plus for Windows* and other PC software.

The complete filename consists of three parts: the main filename, a period (.) and the optional extension, so the format is:

Filename.Ext

The main *Filename* can be 1–8 characters long, and can include any combination of the letters A–Z, numerals 0–9 and certain punctuation marks or other symbols. *Filename* must **not** contain any spaces, commas, question-marks, asterisks (*), backslashes (\) or periods (.).

The period (.) separating the main name and the extension is mandatory.

The extension *Ext* is optional and can be up to three characters, following the same rules as the main *Filename*.

Filenames are not case-sensitive, and any upper/lower-case formatting in your input will be removed by either GRPS or MS-DOS.

Keypad interface

Marksman machines equipped with a keypad and LCD display have access to sub-set of GRPS commands through a system of main menus and sub-menus.

The description of each GRPS command in this manual includes its keypad menu option where available. The general format is:

Keypad menu`Main_menu / Sub_menu`

Error messages are displayed on the LCD as a continuously scrolling ‘banner’. Press any key to clear the error message.

Availability of Keypad Commands

As noted on page vi, there are differences between the GRPS commands available on different models of the *Marksman*: the *Marksman 660*, *Marksman 360* and *Marksman 400/410*.

- **Marksman 660**

Figure 2 beginning on page 19 shows the organizational ‘map’ of the keypad interface menu structure for the *Marksman 660*.

The keypad functions available will depend on the hardware configuration of the machine, so not all of those listed in Figure 2 may be available. Specialized configurations may have additional functions.

- **Marksman 360**

Figure 3 beginning on page 21 shows the organizational ‘map’ of the keypad interface menu structure for the *Marksman 360*.

- **Marksman 400/410**

Figure 4 beginning on page 22 shows the organizational ‘map’ of the keypad interface menu structure for the *Marksman 400* and *410*.

Some commands are available on the *Marksman 410* only.

Figure 2: Keypad interface menus for the *Marksman 660*

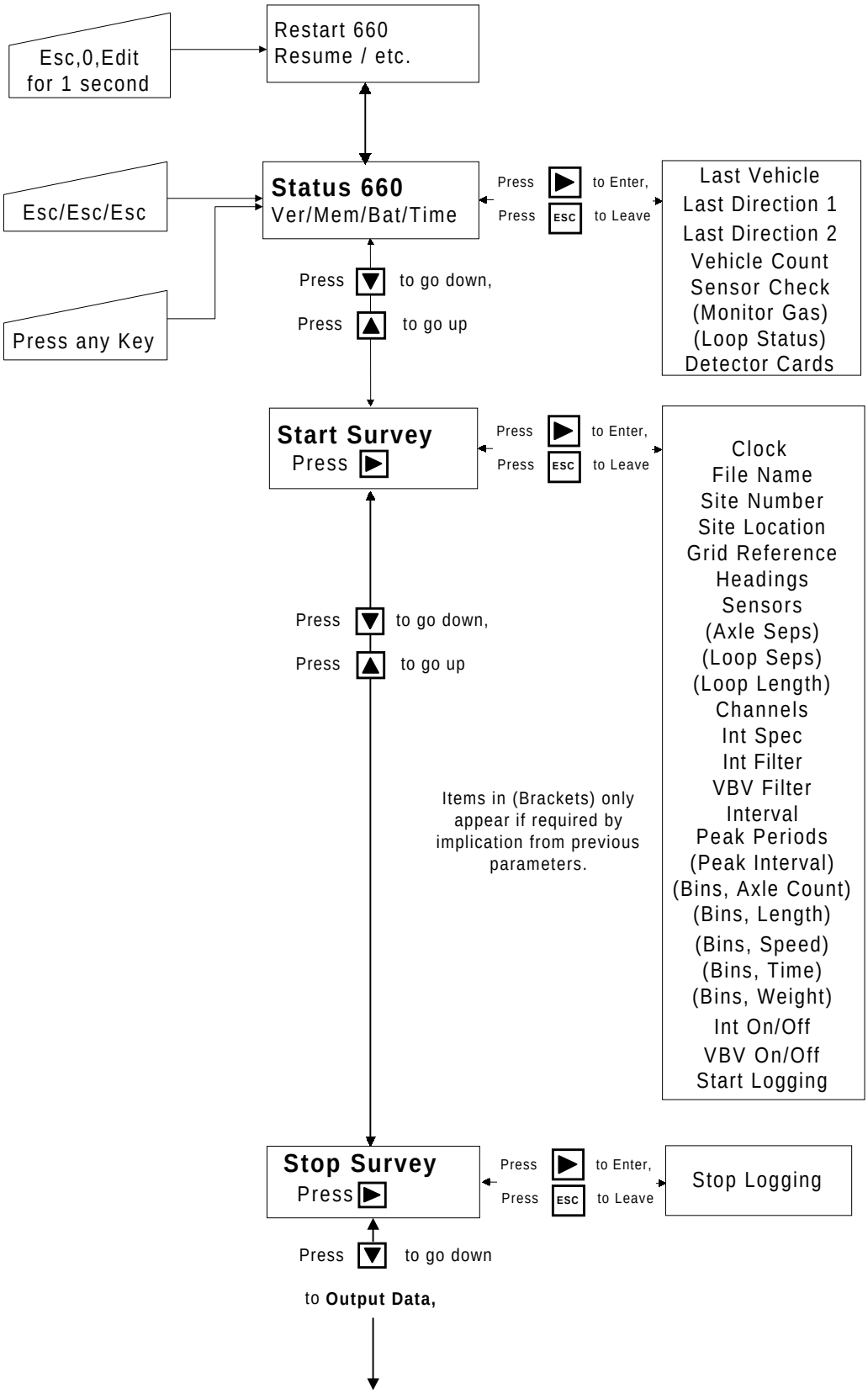


Figure 2: Keypad interface menus for the *Marksman 660* (continued)

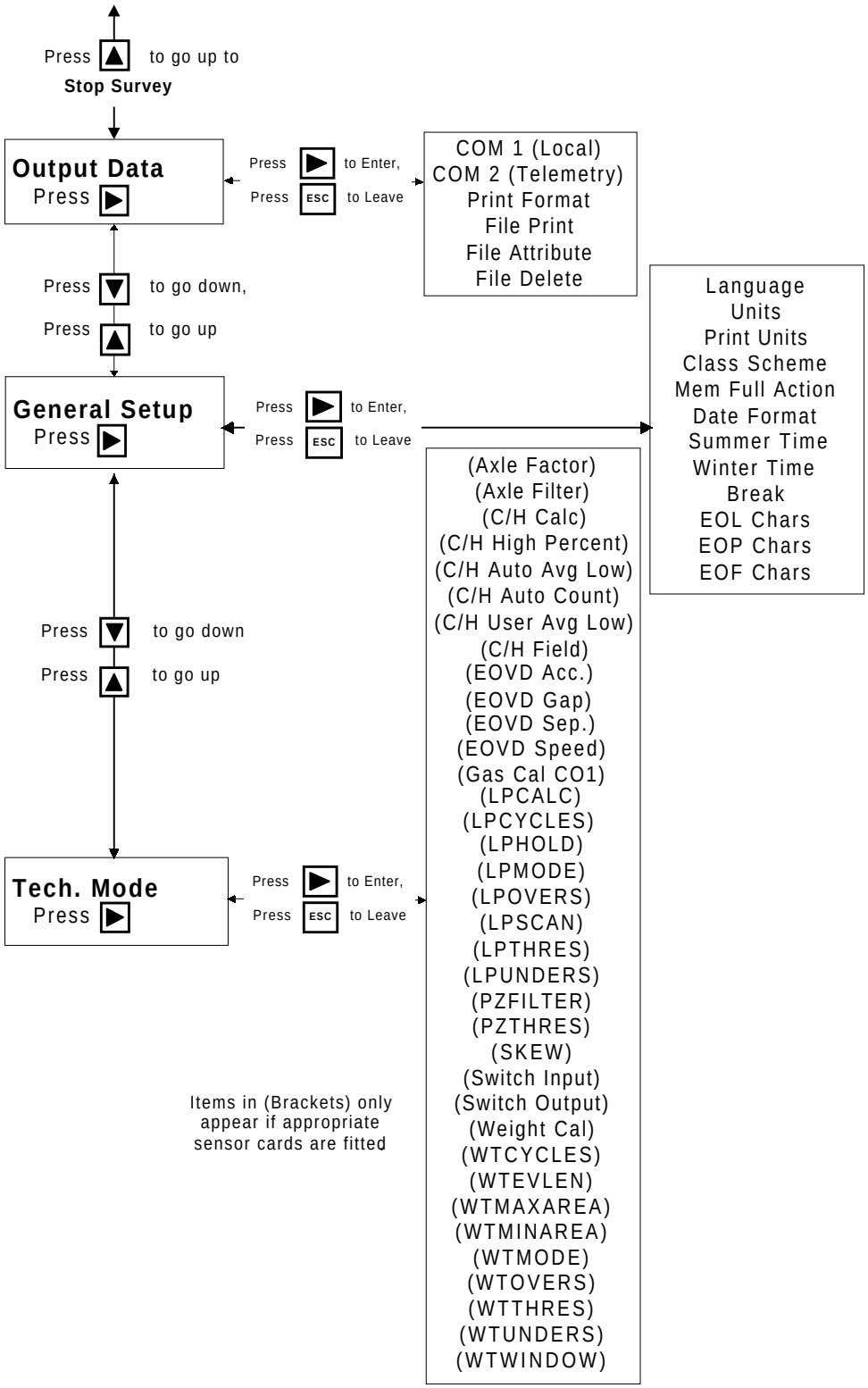


Figure 3: Keypad interface menus for the *Marksman 360*

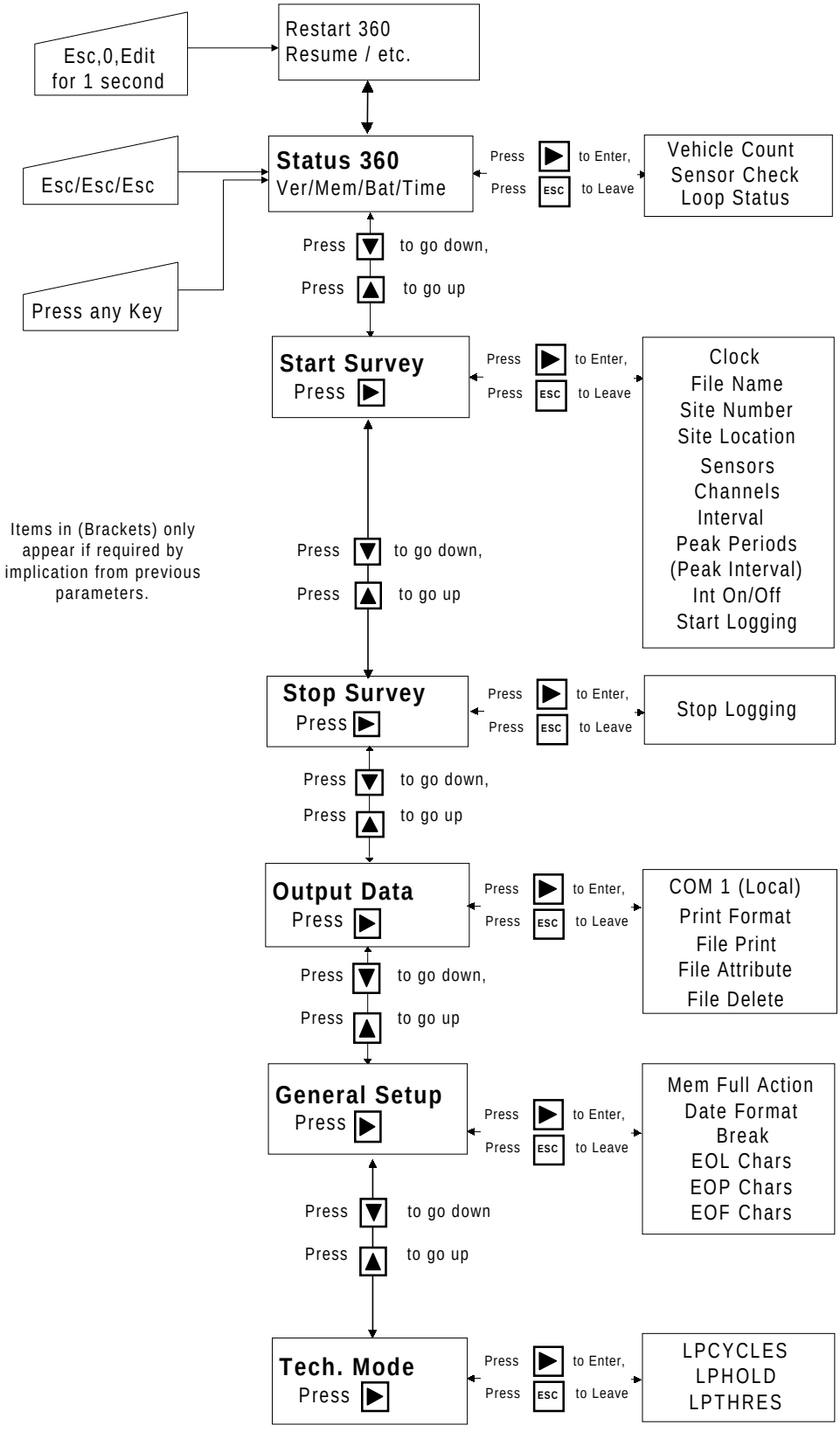
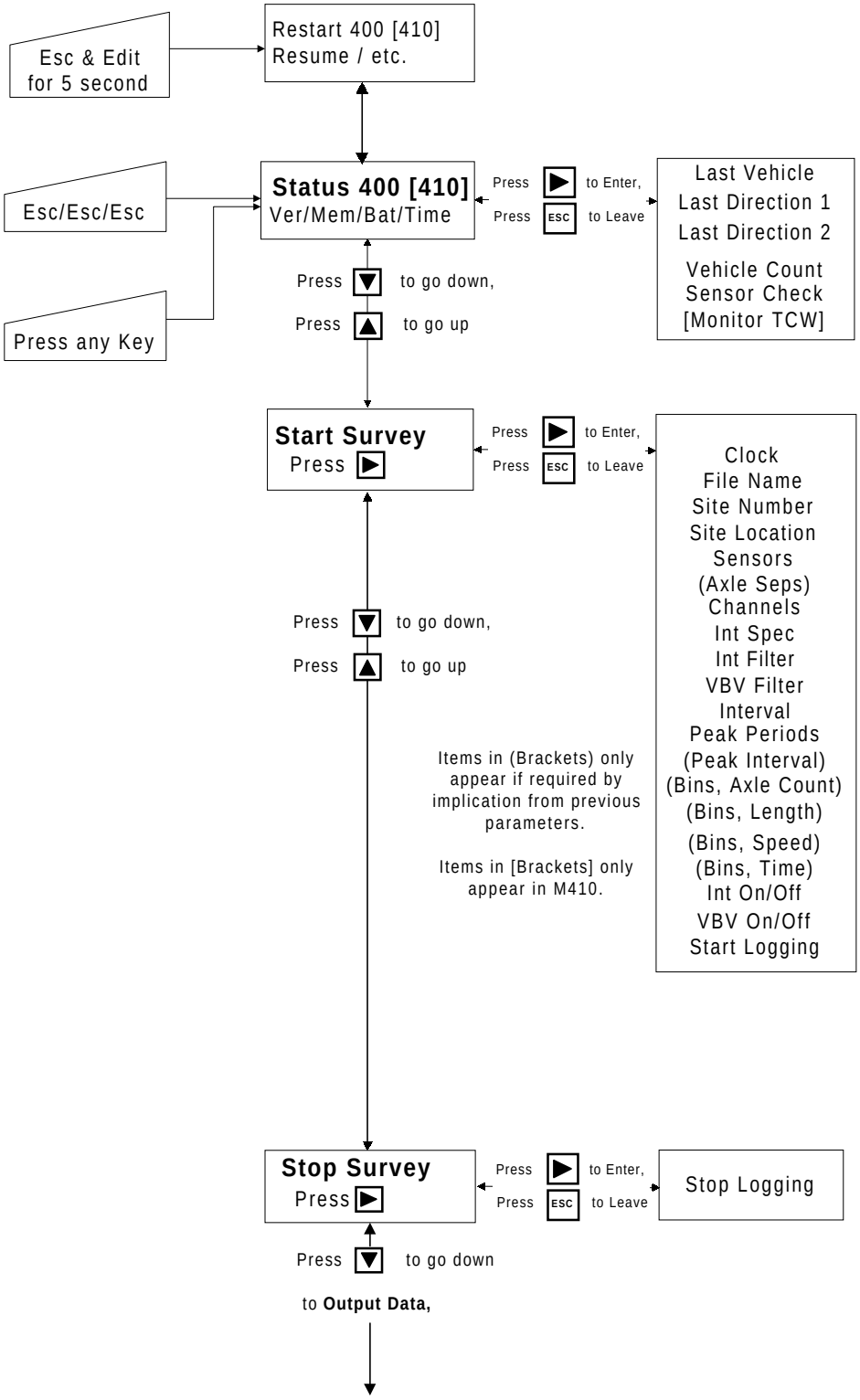
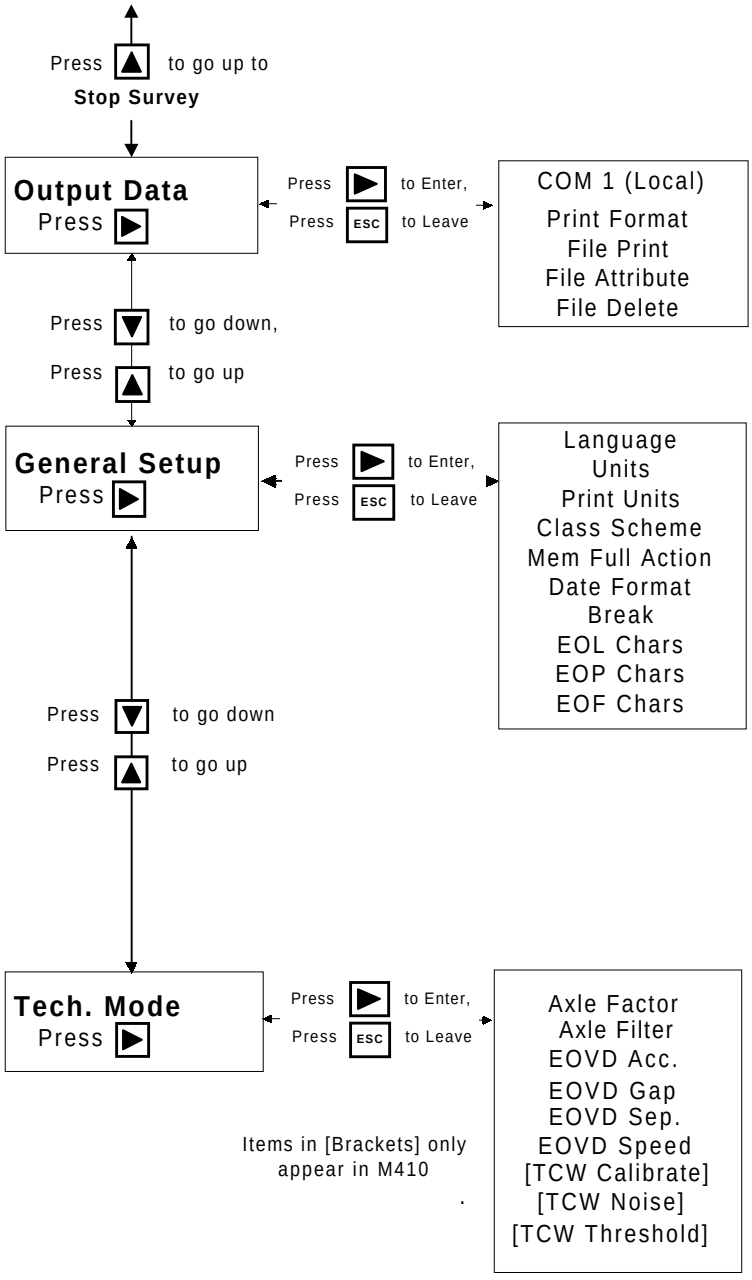


Figure 4: Keypad interface menus for the *Marksman 400* and *Marksman 410*



Some commands are available on the *Marksman 410* only.

Figure 4: Keypad interface menus for the *Marksman 400* and *Marksman 410* (continued)



Some commands are available on the *Marksman 410* only.

GRPS Commands: Alphabetical Reference

This manual covers all GRPS functions and machine configurations. According to the configuration of any particular machine, certain commands may not be available or may have no effect.

This symbol indicates a ‘technical’ command which very few users will ever need. Incorrect use of ‘technical’ commands may result in inaccurate recording or loss of data, so you are advised to consult Golden River or your national Golden River representative before using them.

Where *Availability* is shown below the description of a command (e.g. CLOCK) this indicates when it was first introduced in a newly-released version of the software. If *Availability* is not mentioned, this implies the command has existed since the first release of software, although several command names have changed.

ACTBINS

Axle CounT BINS

Description

In an axle-count configuration, vehicles having different numbers of axles can be grouped together in the same 'bin' for counting purposes. ACTBINS controls this grouping.

This method of classification should be used in a region where there is no standard for vehicle classification. After defining the ACTBINS, use the ACT parameter in the INTSPEC command to enable axle-count classification.

Essential when

INTSPEC includes ACT.

Command syntax

actbins = *LowestBin* [*BinBoundary*...] *HighestBin*

Parameters

LowestBin is the lowest axle count that will be included in the lowest bin (but 0 and 1 have no practical meaning).

BinBoundary is the lowest axle count that will be included in the bin **to the right**.

HighestBin is one axle more than the highest axle count that will be included in any bin.

Valid range

Up to 15 bin boundaries (defining up to 14 bins)

Bin numbers from 0 to 99

Bin numbers must be entered in increasing order.

Default

actbins = 0 2 3 4 5 6 7 8 99

Notes

Except for *lowest bin*, all other bin boundaries mark the beginning of the **next** bin. An axle count equal to that boundary value will be recorded in the bin **below**. Vehicles whose axle count falls outside the stated ranges will not be recorded.

Example	**actbins = 2 3 4 99**
Explanation	The four numbers define three bins. Bin 1 contains vehicles with 2 axles Bin 2 contains vehicles with 3 axles Bin 3 contains vehicles with 4-98 axles
Display syntax	**actbins**
Example response	`ACTBINS = 0 1 2 3 4 5 6 7 8 99`
Keypad menu	`Start Survey / Bins, Axle Count` The display shows the current number of bins.
Related commands	INTSPEC is used to select this type of classification.

Your setting **actbins =** .....

Your notes

ALL (HELP)

Description

ALL lists all the commands in alphabetical order. If a command has variable parameters, the settings are displayed. For commands without variables, a brief line of text beginning with REM explains the function.

ALL is a useful reminder of the format of each command.

To terminate the output before the end, press **Esc** or **Ctrl-C**.

Display syntax **all**

Example response

ALL does not display any parameters related to sensor cards that are not fitted to the machine. For example, the GAS and GASCAL commands will only be displayed if the machine contains a Pollution Monitor card.

```
ACTBINS = 0 2 3 4 5 6 7 8 99
REM ALL,HELP - Displays this information
AXFACTOR = 2.000
AXFILTER = 20
AXSEPS = 400 400 400 400 400 400 400 400
BREAK = DAILY
REM CFGLOAD - Loads a stored configuration
REM CFGSAVE,CFGSTORE - Stores a configuration
CHANNELS = 1 1 1 1 1 1 1 1
```

etc.

etc.

```
WTMODE = 1
WTOVERS = 2
WTTHRES = 8
WTUNDERS = 2
WTWINDOW = 16
```

You are then returned to the current GRPS input prompt.

Keypad menu Not available

Related commands HELP is an alternative command name for ALL.

PAGELEN sets the number of lines of text, after which the output pauses with a prompt of `------ More ------`. Press **Enter** to continue.

REM is a comment line which is ignored.

STATUS lists the machine's hardware status.

AXFACTOR

AXle count calibration FACTOR

Description	This command affects all sensors when using a count configuration. The axles count is divided by AXFACTOR to give an estimated vehicle count. AXFACTOR is therefore an estimated average number of axles per vehicle.
Essential when	using any count configuration, i.e. whenever INTSPEC contains CNT.

Command syntax	**axfactor =** *Divisor*
Parameters	*Divisor* is the number by which the axle count is divided to obtain the vehicle count when the data file is printed. The actual axle count data are unaffected.
Valid range	0.10 to 9.99 axles per vehicle
Default	axfactor = 2.00
Example *Explanation*	**axfactor = 2.40** If 2400 axle counts are registered in a recording period, a vehicle count of 1000 would be displayed when the data file is printed.
Display syntax	**axfactor**
Example response	`AXFACTOR = 2.00`
Keypad menu	`Tech Mode / Axle Factor`

Your setting **axfactor =** .....

Your notes

AXFILTER

Axle detector FILTER

Description Axle detectors typically do not produce one pulse when the axle passes over, but several. To 'debounce' this signal, AXFILTER defines a 'dead time' after the first pulse, during which any further pulses generated by the same sensor will be ignored.

For piezo detectors, use the PZFILTER command which programs the debounce time directly into the detector card.

Essential when tube counting layouts are in use.

Command syntax **axfilter =** *Time*

Parameters *Time* is the dead-time in milliseconds.

Valid range 1 to 999 milliseconds

Default axfilter = 20 [General] = 30 [M410]

Example **axfilter = 35**

Display syntax **axfilter**

Example response `AXFILTER = 25`

Keypad menu `Tech Mode / Axle Filter`

Related commands With piezo detectors, use PZFILTER to program the debounce time directly into the detector card.

Your setting **axfilter =** .....

Your notes

AXSEPS

AXle detector SEParationS

Description	When two (or more) axle sensors of the same type are installed in each lane, AXSEPS informs the machine of the separation between them in order to calculate the vehicle speed.
Essential when	the SENSORS specification involves two axle sensors per lane, e.g. TT or PP.

Command syntax **axseps =** *Lane1* [*Lane2...*]

Parameters

Lane1 is the axle detector separation in lane 1, in the units defined using the UNITS command. If no other parameters are given, the same value is applied to all lanes defined by SENSORS.

Lane2... are the values for lane 2, and beyond if applicable. Up to 8 lanes are allowed.

Valid range 1 to 999 centimetres or 1 to 393 inches.

Default axseps = 400 cm or axseps = 157 inches
This value applies to all lanes (or to the only lane).

Notes Lane 1 is nearest to the machine location.
Once set, the AXSEPS value will be converted automatically if you change UNITS (subject to rounding).

Example 1 **axseps = 600**
Explanation This value is only valid in conjunction with units = metric. The distance between axle sensors in all lanes (or the only lane) is 600 cm.

Example 2 **axseps = 95 100**
Explanation Assuming metric units, the distance between axle sensors is 95 cm in lane1 and 100 cm in lane 2.

Display syntax	**axseps**
Example response	`AXSEPS = 500 500 500 500 500 500 500 500`
Keypad menu	`Start Survey / Axle Seps`
Related commands	UNITS sets the length units to centimetres or inches. SENSORS defines the sensor layout.

Your setting **axseps =** .....

Your notes

BREAK

Description

BREAK provide the means to break recording files at regular intervals, for example once a day. At the break, all open files are closed. A new file is opened, with the same name but the file extension is incremented by 1.

BREAK is useful when large data files are expected. Smaller files are easier to handle, and limit the risks if technical problems are affecting recordings.

Essential when

users who have **Read-only** access to the machine via the remote serial link (see INTERFACE) need to retrieve all available data. Using BREAK to divide the data into several successive files gives such users access to all data except the most recent in the currently open file.

Command syntax **break =** [**hourly / daily / weekly / off**]

Parameters

An **hourly** file break will be inserted at the end of each hour, a **daily** file break will occur at midnight and a **weekly** file break at midnight on Sunday night.

off will cancel any previously set break interval.

Default

break = off

Notes

The BREAK command automatically opens new files, so take care that the number or size of the files to be stored will not exceed the maximum allowed by the machine's memory size. If the number or size of the files to be created will exceed the maximum allowed, the action of BREAK will be determined by the setting of MEMFULL.

Example

break = hourly

Explanation

24 files will be created per day

Display syntax **break**

Example response

```
BREAK = Off
```

Explanation

The BREAK function is currently not enabled.

Keypad menu `General Setup / Break`

Related commands MEMFULL will determine the action of BREAK if the number or size of the files to be created will exceed the maximum allowed.

STARTREC starts a recording immediately.

STOPREC stops a recording immediately.

INTONOFF and VBVONOFF will set the starting and stopping times of recordings in advance.

Your setting **break =** .....

Your notes

CFGLOAD

ConFiGuration LOAD

Description

This command loads and implements a previously saved machine configuration, including all GRPS parameters except TIME and DATE.

CFGLOAD can be used to 'clone' one machine's configuration into several others. A configuration file is created in the first machine's memory using CFGSAVE, and then downloaded into a PC using RETRIEVE. Then UPLOAD the configuration file from the PC into each machine, and then using CFGLOAD to put that configuration into effect.

Command syntax **cfgload** *Filename*

Parameters *Filename* is a valid MS-DOS/GRPS filename (up to 8 characters). Do not include an extension after the filename.

Notes The file specified by *Filename* must be present in the machine's memory. The machine generates an extension to the filename in the series beginning **.c00**.

Example **cfgload file1234**
Explanation Implements all the configuration settings contained in that file, except for COM1, COM2, TIME and DATE

Keypad menu Not available

Related commands ALL lists all the machine's current settings.

CFGSAVE saves the current configuration to a file in the machine's memory.

DIR lists all files in the machine's memory.

UPLOAD will upload a new configuration file from a PC.

Your setting **cfgload =** (*filename*).....

Your notes

CFGSAVE (CFGSTORE)

ConFiGuration SAVE

Description

This command saves the machine's configuration to a file in memory, including all GRPS parameters except TIME and DATE.

The Description of CFGLOAD explains how to use CFGSAVE in 'cloning' one machine's configuration into several others.

Command syntax **cfgsave** *Filename*

Parameters

Filename is a valid MS-DOS/GRPS filename (up to 8 characters). Do not include an extension after the filename.

If no *Filename* parameter is supplied, the one already stored using the FILENAME command will be used.

Notes

If the *Filename* specified already exists in the machine's memory (or there is a configuration file corresponding to the current FILENAME setting), that file will be overwritten – use the DIR command first to avoid this.

Example 1
cfgsave file1234
Explanation
Saves all the current configuration settings to that file, except for TIME and DATE

Example 2
cfgsave
Explanation
If the current setting of FILENAME is 'file1234', this will save all the current configuration settings except TIME and DATE to a file called **file1234.c*nn***, where '*nn'* is an automatically generated numerical extension starting from **c00**.

Keypad menu Not available

Related commands

ALL lists all the machine's current settings.

CFGLOAD will load and implement a configuration file from the machine's memory.

DIR lists all files in the machine's memory.

UPLOAD will upload a new configuration file from a PC.

Your notes

CHANNELS

Description

Interval recordings usually accumulate all lanes of traffic into one set of data. CHANNELS allows the assignment of traffic lanes into two or more independent channels, each of which accumulates data separately.

CHANNELS is rarely required, except to change from the default setting in the *Marksman*. If two directions of data are required, e.g. for the two sides of a divided highway, use INTSPEC and the DRN parameter in preference to CHANNELS.

Command syntax **channels =** *N1* [*N1*...] [*N2* [*N2*...]] etc.

Parameters

N1 is the channel number to which the first lane is assigned. If there are no other parameters, all lanes are assigned to channel *N1*.
If there is more than one parameter, lanes will be assigned in sequence to the channel numbers given.

Valid range

1 to 1 2 3 4 5 6 7 8
There is a maximum of 8 lanes or 8 channels, whichever is reached first.
A space is required between numbers.
Channel numbering must be continuous, with no missing channels; for example 1 1 3 3 is invalid.

Default

channels = 1 1 1 1 1 1 1 1

Notes

A different format of PRINT occurs when CHANNELS is selected for two channels.

Example 1

channels = 1

Explanation

Data from all lanes will go into channel 1.

Example 2 **channels = 1 2**
Explanation Data from the first lane will go into channel 1, and data from the second lane into channel 2.

➲ **CAUTION**

When the CHANNELS default setting is changed to use more than one parameter as shown in the example above, data from lanes that are not assigned a channel will be ignored.

In the above example no data would be recorded for lanes 3, 4, 5, 6, 7 or 8.

Display syntax **channels**

Example response `channels = 1 1 1 1 1 1 1 1`

Keypad menu `Start Survey / Channels`

Related commands INTSPEC specifies how interval data are to be recorded. The INTSPEC parameter DRN is an alternative to CHANNELS if data are to be collected by direction.

Your setting **channels =** .....

Your notes

CHAUTAVLOW

Chassis Height AUTomatic AVeraging, LOW

Description	CHAUTAVLOW is a display-only command for sensor 'tuning', for use with the loop-loop classification schemes only. It is a new name for LPCLS2. CHAUTAVLOW displays the running-average change in peak sensor output for the last 100 cars. To calibrate the chassis height sensing which is used to determine 'low' or 'high' chassis, the CHAUTAVLOW value is compared against the threshold between 'high chassis' and 'low chassis' which is set using CHHIGH%.
Availability	Version 1.94 onwards (from version 1.9 as LPCLS2).
Essential when	calibrating a loop-loop sensor system for classification.

Display syntax	**chautavlow**
Example response **Explanation**	`CHAUTAVLOW = 182 174 160 190 0 0 0 0` One value for each lane in use, up to eight lanes
Note	The running average result from CHAUTAVLOW is not valid until CHAUTCNT verifies that at least 100 cars have been counted.
Keypad menu	`Tech Mode / C/H Auto Avg Low`
Related commands	LPCLS2 is the previous name for this command, and is currently still valid. CLASS selects the loop-loop classification scheme (e.g. EUR6). CHAUTCNT (CLS2CNT) verifies that at least 100 cars have been counted. CHHIGH% (LP2THRES) is the associated sensor threshold command.

Your notes

CHAUTCNT

Chassis Height AUTo CouNT

Description	CHAUTCNT is a display-only command for sensor 'tuning', for use only with loop-loop classification schemes that are based on chassis height (e.g. EUR6). It is a new name for CLS2CNT. CHAUTCNT displays the number of vehicles that have been detected as cars for each lane. This value needs to be at least 100 in order to give reliable results from readouts using the CHAUTAVLOW command. The count value in each lane is set to zero when the survey commences, and the maximum value ever displayed is 100.
Availability	Version 1.98 onwards.
Essential when	calibrating a loop-loop sensor system for classification.

Display syntax	**chautcnt**
Example response 1	`CHAUTCNT = 42 100 0 0 0 0 0 0`
Explanation	Insufficient vehicles counted for lane 1; wait until display shows `100`.
Example response 2	`CHAUTCNT = 100 100 0 0 0 0 0 0`
Explanation	Count is at least 100 in both lanes being monitored, so CHAUTAVLOW values will be valid.
Keypad menu	`Tech Mode / C/H Auto Count`
Related commands	CLS2CNT is the earlier name for this command, and is currently still valid. CHAUTAVLOW displays the running-average signal levels for the last 100 cars.

CHCALC defines the method of calculating the average height of a ‘low’ chassis (automatic or user input).

CHFIELD can be used to correct the length measurements made by loop sensors.

CHHIGH% sets or displays the threshold value between detection signals that are counted as a ‘low chassis’ and those counted as a ‘high chassis’.

CHUSRAVLOW allows the user to enter the loop signal level corresponding to a ‘standard’ car with a low chassis height.

CLASS selects the loop-loop classification scheme (e.g. EUR6).

Your setting **chautcnt =** .....

Your notes

CHCALC

Chassis Height CALCulation method

Description	For vehicle classification using loop sensors, the *Marksman 660* determines the height of the chassis by comparing the maximum loop output signal against the value for a 'standard car', derived from a test run. CHCALC controls how the value for the standard car is calculated.
Availability	Version 1.98 onwards.
Essential when	calibrating a loop-loop sensor system for classification based on chassis height.

Command syntax	**chcalc = auto / user**
Parameters	**auto** makes the *Marksman 660* calculate the loop output value for a standard car. Use the **auto** mode for LL sensor configurations. **user** will use the loop value entered using the CHUSRAVLOW command. Use the **user** mode for sensor configurations 2(N+1)*2, 2(N+1)*3, 2(N+1)*4 and 2(N+2)*4.
Default	chcalc = auto
Display syntax	**chcalc**
Example response	`CHCALC = User`
Keypad menu	`Tech Mode / C/H Calc`
Related commands	CHAUTAVLOW displays the running-average signal levels for the last 100 cars, and CHAUTCNT verifies that sufficient vehicles have been counted. CHFIELD can be used to correct the length measurements made by loop sensors. CHHIGH% sets or displays the threshold value between detection signals that are counted as a 'low chassis' and those counted as a 'high chassis'.

CHUSRAVLOW allows the user to enter the loop signal level corresponding to a ‘standard’ car with a low chassis height.

CLASS selects the loop-loop classification scheme (e.g. EUR6).

Your setting **chcalc =** .....

Your notes

CHFIELD

Chassis Height loop detection FIELD

Description	During vehicle classification using loop sensors, the lengths of vehicles with a very low chassis height tend to be overestimated, while the lengths of vehicles with high chassis tend to be underestimated. CHFIELD can compensate for this effect by decreasing the measured length of a vehicle identified as having a low chassis, and increasing the measured length of a vehicle identified as having a high chassis.
Availability	Version 1.98 onwards.
Essential when	calibrating a loop-loop sensor system for classification based on chassis height.

Command syntax	**chfield =** *MaxLength [MaxLength2...]*
Parameters	*MaxLength* is the maximum allowable upward or downward length adjustment. If only a single value is given, it will be applied to all sensors in the array. If multiple values are given (*MaxLength2* etc.) there must be a value for each loop in the array. Adjust the CHFIELD values to give correct length readings for vehicles passing over the loops.
Valid range	0 to 999 centimetres or 0 to 373 inches, according to the current setting of UNITS.
Default	chfield = 0
Example *Explanation*	**chfield = 100** UNITS are centimetres.
Display syntax	**chfield**
Example response	`CHFIELD = 80`
Keypad menu	`Tech Mode / C/H Field`

Related commands

CHAUTAVLOW displays the running-average signal levels for the last 100 cars, and CHAUTCNT verifies that sufficient vehicles have been counted.

CHCALC defines the method of calculating the average height of a ‘low’ chassis (automatic or user input).

CHHIGH% sets or displays the threshold value between detection signals that are counted as a ‘low chassis’ and those counted as a ‘high chassis’.

CHUSRAVLOW allows the user to enter the loop signal level corresponding to a ‘standard’ car with a low chassis height.

CLASS selects the loop-loop classification scheme (e.g. EUR6).

Your setting **chfield =** .....

Your notes

CHHIGH%

Chassis Height HIGH, percentage

Description

CHHIGH% is a sensor 'tuning' command, for use with loop-loop classification schemes only (e.g. EUR6). It is a new name for LP2THRES.

CHHIGH% sets or displays the threshold value between detection signals that are counted as a 'low chassis' and those counted as a 'high chassis'.

This setting must be adjusted to suit the site and typical vehicle types. To assist in this, CHAUTAVLOW displays the running-average signal levels for the last 100 cars, and CHAUTCNT verifies that sufficient vehicles have been counted.

Availability Version 1.98 onwards.

Essential when calibrating a loop-loop sensor system for classification.

Command syntax **chhigh% =** *Threshold* / [*Threshold...*]

Parameters *Threshold* is the running-average change in CHAUTAVLOW, expressed as a percentage.
A single value applies to all lanes, but separate values can be set for up to eight lanes in sequence.

Note If the signal from a particular vehicle is higher than (CHAUTAVLOW x CHHIGH%), that vehicle will be counted as a 'low' chassis. Therefore if the system is counting some high-chassis vehicles as 'low-chassis', increase CHHIGH%; if the system is counting some low-chassis vehicles as 'high-chassis', decrease CHHIGH%.

Valid range 1 to 100

Default chhigh% = 50 50 50 50 50 50 50 50

Example **chhigh% = 45**
Explanation Same value for all lanes (or else there is only one lane)

Display syntax **chhigh%**

Example response	`CHHIGH% = 65 63 50 50 50 50 50 50`
Explanation	Two lanes have been altered; the rest are still at the default value of 50
Keypad menu	`Tech Mode / C/H HIGH PERCENT`
Related commands	CHAUTAVLOW displays the running-average signal levels for the last 100 cars, and CHAUTCNT verifies that sufficient vehicles have been counted. CHCALC defines the method of calculating the average height of a 'low' chassis (automatic or user input). CHFIELD can be used to correct the length measurements made by loop sensors. CHUSRAVLOW allows the user to enter the loop signal level corresponding to a 'standard' car with a low chassis height. CLASS selects the loop-loop classification scheme (e.g. EUR6).

Your setting **chhigh% =** .....

Your notes

CHMOD

CHange file MODe

Description Data files held in the memory of the *Marksman* series have a 'mode' attribute which indicates their status as Open (O), Retrieved (R) or Un-retrieved (U). The CHMOD command changes the attribute of one or more files to 'R' or 'U', provided that the files are not currently Open.

Essential when deleting a file whose present attribute is Un-retrieved (U); use CHMOD R first to change the file's attribute to 'R'.

CHMOD U useful if retrieved copies of files have been lost.

Command syntax **chmod r / u** *Filename.Ext* / **all**

Parameters The attribute **r** or **u** is required, and will become the new attribute of the chosen file(s).

Filename.Ext is a valid MS-DOS/GRPS filename (up to 8 characters) followed by a period (.) and an extension of up to three characters.

all will change all files that are not currently Open (O).

Valid range The selected file(s) must not currently be Open (O).

Example 1 **chmod r file1234.i01**
Explanation The file FILE1234.I01 will be marked as Retrieved (R) unless it is currently Open (O).

Example 2 **chmod u all**
Explanation All files that are not currently Open (O) will be marked as Un-retrieved (U).

Keypad menu `Output Data / File Attribute`

Related commands DELETE is used to delete files after they have been retrieved from the machine's memory, and for safety will only operate on files marked as Retrieved (R).

DIR lists the attributes of all files.

PRINT is used to display a file in the machine's memory, and when completed will mark the file as Retrieved (R).

RETRIEVE is used to retrieve files from the machine's memory, and when completed will mark each file as Retrieved (R).

STOPREC ends recording, closes the open data file and changes its attribute from Open (O) to Un-retrieved (U).

Your notes

CHUSRAVLOW

Chassis Height USeR defined AVerage LOW chassis

Description	For vehicle classification based on chassis height measured using loop sensors, CHUSRAVLOW allows the user to enter the loop signal level corresponding to a 'standard' car with a low chassis height. CHUSRAVLOW is only applicable if you have already set **chcalc = user**. If you set **chcalc = auto**, the *Marksman 660* will make its own determination.
Availability	Version 1.98 onwards.
Essential when	calibrating a loop-loop sensor system for classification based on chassis height.

Command syntax	**chusravlow =** *Value [Value2...]*
Parameters	*Value* is the result that was obtained by driving a 'standard' car exactly over the loop sensor with the OSP+ printout enabled. The required value is the one in the PEAK column of the printout, associated with the loop OFF event (Appendix 2). If only a single value is given, it will be applied to all sensors in the array. If multiple values are given (*Value2* etc.) there must be a value for each loop in the array.
Valid range	1 to 60000 (in the internal units used by OSP+).
Default	chusravlow = 187
Example	**chusravlow = 220**
Display syntax	**chusravlow**
Example response	`CHUSRAVLOW = 250`
Keypad menu	`Tech Mode / C/H User Avg Low`
Related commands	CHAUTAVLOW displays the running-average signal levels for the last 100 cars, and CHAUTCNT verifies that sufficient vehicles have been counted.

CHCALC defines the method of calculating the average height of a ‘low’ chassis (automatic or user input).

CHFIELD can be used to correct the length measurements made by loop sensors.

CHHIGH% sets or displays the threshold value between detection signals that are counted as a ‘low chassis’ and those counted as a ‘high chassis’.

CLASS selects the loop-loop classification scheme (e.g. EUR6).

Your setting **chusravlow =** .....

Your notes

CLASS

Vehicle CLASSification scheme

See page 60 for *Marksman 410* enhancements.

Description Most countries have a standard Vehicle Type Classification scheme. For example EUR6 is a common scheme in Europe, and FHWA13 is used extensively in North America.

The *Marksman* instruments have a built-in range of vehicle class schemes, and CLASS selects the one to be used for both vehicle-by-vehicle and interval recording.

Essential when INTSPEC, INTFILTER and/or VBVFILTER contain CLS. CLS is only available when wheelbase information is to be collected (except for EUR6 which uses chassis height information from two-loop sensors).

Command syntax **class =** *Scheme*

Parameters *Scheme* must be one of the following:

asra13	Australian 13-class (1986)
aust13	Australian 13-class (1994)
cal15	Californian 15-class
eur6	European 6-class
eur13	European 13-class
fhwa13	Federal Highway 13-class
fhwa15	Federal Highway 15-class
stg11	UK Statistics Group 11-class
swed13	Swedish 13-class
swiss7	Swiss 7-class
tnz13	Transit New Zealand 13-class
wmjdt6	West Midlands Joint Data Team 6-class

Default class = eur13

Notes	CLASS is never reset automatically to the default, even by a RESTART 4 command; any change must be entered manually.
Example	**class = fhwa13**
Display syntax	**class**
Example response	`CLASS = SWED13`
Keypad menu	`General Setup / Class Scheme`
Related commands	INTSPEC selects when class data are to be collected. INTFILTER and VBVFILTER may both use CLASS to select vehicles for recording. CLS2CNT, LPCLS2, LP2THRES are sensor 'tuning' commands for EUR6 only. See page 60 for *Marksman 410* enhancements.

Your setting **class =** .....

Your notes

CLASS (*Marksman 410* enhancements)

Vehicle Classification Scheme

Description The CLASS command for the *Marksman 410* has extra options for bicycle classification schemes.

If one of the original *Marksman 660* classification schemes (page 58) is selected when using the *Marksman 410*, bicycles will be classified in the same class as motorcycles.

Note Only the *Marksman 410* enhancements are detailed below. For the remaining details of the CLASS command, see page 6658.

Command syntax **class =** *Scheme*

Parameters In addition to all the options for *Scheme* detailed on page 58, the *Marksman 410* also allows one of the following:

bic2	*2-class Bicycle Scheme*
nl6	*Netherlands 6-class Bicycle Scheme*
nlb13	*Netherlands 13-class Bicycle Scheme*

For further details of these schemes, see opposite.

Default class = eur13

Example **class = bic2**

Your Setting **class =** ……

BIC2

This is a simple classification scheme which has two classes, one for bicycles and one for motor vehicles.

The BIC2 scheme can only be used with any of the following sensor configurations: *T, T*2, TT, TT*2.*

NL6

The Netherlands 6-class scheme is based on the EUR13 classification scheme, with a class for bicycles.

Class	Vehicles	EUR 13 equiv
1	Car, Car with Trailer, Van or LGV	1
2	Rigid Truck	2, 3, 4
3	Rigid Truck with Trailer or Articulated Lorry	5–11
4	Bus or Coach	12
5	Bicycle	none
6	All other vehicles	13

The NL6 scheme can only be used with the following sensor configurations: *TT, TT*2.*

NLB13

The Netherlands 13-class scheme is again based on the EUR13 classification scheme, with a class for bicycles.

Class	Vehicles	EUR 13 equiv
1–9	As EUR13	1–9
10	Articulated Lorry with 3-axle Tractor and 3-axle Semi-Trailer	10, 11
11	Bicycle	none
12, 13	As EUR13	12, 13

The NLB13 scheme can only be used with the following sensor configurations: *TT, TT*2.*

CLOCK

Set CLOCK and calendar

Description

This command resets the *Marksman*'s internal clock/ calendar, changing the time of day and the date in a single operation.

CLOCK supersedes the separate TIME and DATE commands, as it gives more reliability for remote operation.

If a survey is running, CLOCK will close all open files while the time and date are being reset. Recording will then resume, opening new continuation files if either the time or the date is different from before. This is consistent with other GRPS commands such as PRINT, RETRIEVE, SUMMER and WINTER.

Availability

Version 1.95 onwards.

Essential when

the clock/calendar needs to be reset (e.g. because both the main battery and the backup battery have been discharged or disconnected) or when the *Marksman*'s clock/calendar needs to be synchronized with that of a remote base station.

Command syntax

clock = *hh:mm:ss Date*

Parameters

hh:mm:ss is the time of day. *hh* are the hours in 24-hour clock format, *mm* are the minutes and *ss* the seconds. The colons are required. Include leading zeros and omit spaces within *hh:mm:ss*.

Include a space between the time and the *Date*.

Date must be in the format specified by the current DATEFORM setting. All formats require '/' as the day/month/year separator. Include leading zeros and omit spaces within *Date*.

Starting with *Marksman* firmware version 1.94, two-digit year numbers must be 96 or later, in order to be

interpreted correctly (as 1996 to 2095). The year can also be entered in four-digit format.

Default — Set to the current time and date when the unit is first powered-up in the factory. CLOCK is not affected by any resets (not even RESTART 4); all changes must be entered explicitly.

Example — **clock = 12:00:00 13/03/98**

Explanation — DATEFORM must be 'DD/MM/YY' for this date to be valid. Note the leading zero in '03'.

Display syntax — **clock**

Example response — `CLOCK = 14:43:54 26/10/98`

Keypad menu — Not yet available – use `Start Survey / Time` and `Start Survey / Date`.

Related commands — TIME sets the time of day, separately from the date. DATE sets the date separately from the time of day. These two commands may be phased out and replaced by CLOCK.

DATEFORM sets the format for the date.

Your notes

CLS2CNT (CHAUTCNT)

CLaSs 2 CouNT

Description	CLS2CNT is a sensor ‘tuning’ command, for use with the loop-loop classification schemes only (e.g. EUR6). CLS2CNT displays the number of vehicles that have been detected as cars for each lane. This value needs to be at least 100 in order to give reliable results from LPCLS2. The count value in each lane is set to zero when the survey commences, and the maximum value ever displayed is 100.
Availability	Version 1.9 onwards.
Essential when	calibrating a loop-loop sensor system for classification.

Display syntax	**cls2cnt**
Example response 1 *Explanation*	`CLS2CNT = 42 100 0 0 0 0 0 0` Insufficient vehicles counted for lane 1; wait until display shows `100`.
Example response 2 *Explanation*	`CLS2CNT = 100 100 0 0 0 0 0 0` Count is at least 100 in both lanes being monitored, so LPCLS2 values will be valid.
Keypad menu	`Tech Mode / C/H Auto Count`
Related commands	CHAUTCNT is an alternative name for this command, in the Chassis Height series beginning with ‘CH’. CLASS selects the loop-loop classification scheme (e.g. EUR6). LPCLS2 gives the running average of peak sensor readings from vehicles identified as cars. LP2THRES is the associated sensor threshold ‘tuning’ command.

Your setting **cls2cnt =** .....

Your notes

COM1, COM2

COMmunications channel 1, 2

Description

COM1 and COM2 commands have identical syntax, and control the communications parameters of the two serial ports: baud rate, bits, parity, and flow-control.

COM1 is the 'Local' serial port for communication with an attached PC Terminal or Data Module.

COM2 (if fitted) is the 'Telemetry' port for attachment of a modem.

The following examples mainly use the COM1 command but apply equally to COM2.

Essential when

preparing to output a keyboard-controlled PRINT file.

Command syntax

com1 = *Speed BitsParity Handshaking*

Parameters

Speed is a baud rate chosen from must be one of the following: *300*, *600*, *1200*, *2400*, *4800*, *9600*, *19200*, *38400*, *57600* or *115200*.

BitsParity must be one of the following combinations:

8n	8 bits, no parity
8e	8 bits, even parity
8o	8 bits, odd parity
7n	7 bits, no parity
7e	7 bits, even parity
7o	7 bits, odd parity

Handshaking is the method of transmit/receive flow control, and must be one of the following:

hh	Hardware flow control (by the RTS/CTS lines)
sh	Software flow control (by XON/XOFF codes)
both	Both hardware and software flow control
off	No flow control

Defaults

com1 = *9600 8n off*

com2 = *1200 8n off*

COM1 and COM2 are never reset automatically to the default, even by a RESTART 4 command; any change must be entered manually.

Notes

Hardware handshaking using the RTS and CTS lines in the serial cable connection affects data transfers using the RETRIEVE and UPLOAD commands.

However, RETRIEVE and UPLOAD operations ignore software handshaking because the file transfer PROTOCOL setting takes precedence.

On receiving a valid COM1 or COM2 command, the machine responds:

```
Alter COM1 equipment serial parameters now
```

Example

com1 = 19200 8n hh

Display syntax

com1

Example response

```
COM1 = 9600 8N OFF
```

Keypad menu

```
Output Data / COM1 (Local)
Output Data / COM2 (Telemetry)
```

Your settings

com1 = ...

com2 = ...

DATE

Description This command will change or display the date held in the machine's internal clock/calendar; but CLOCK is now the preferred command to do this.

Essential when the clock/calendar needs to be reset (e.g. because both the main battery and the backup battery have been disconnected).

Command syntax **date =** *Date*

Parameters *Date* must be given in the format specified by the current DATEFORM setting.

All formats require '/' as the day/month/year separator; include leading zeros and omit spaces.

Valid range 01/01/00 to 31/12/99 (or equivalent in other DATEFORM formats). Year entries in two-digit format beyond AD2000 are interpreted correctly, and the year can also be entered in four-digit format.

Default Set to the current date when the unit is first powered-up in the factory. DATE is not affected by any resets (not even RESTART 4); all changes must be entered explicitly.

Example **date = 30/09/98**
Explanation DATEFORM must be 'DD/MM/YY'.
Note the leading zero in the month 09.

Display syntax **date**

Example response `DATE = 30/09/98`

Keypad menu `Start Survey / Clock`

Related commands CLOCK sets both the date and time of day, and should be used in preference to DATE and TIME separately.

DATEFORM sets the format for the date.

TIME sets the time of day.

DATEFORM

DATE FORMat

Description Display or change the date format.

Command syntax **dateform = dd/mm/yy / mm/dd/yy / yy/mm/dd**

Parameters Enter one of the above formats. Subsequent date displays will use two digits each for the day (**dd**), month (**mm**) and year (**yy**).

Default dateform = dd/mm/yy
DATEFORM is not affected by any resets (not even RESTART 4); all changes must be entered explicitly.

Example **dateform = yy/mm/dd**

Display syntax **dateform**

Example response `DATEFORM = YY/MM/DD`

Keypad menu `General Setup / Date Format`

Related commands DATEFORM affects the output of every command that displays the date.

CLOCK (preferred to DATE) displays or changes the date, using the format specified using DATEFORM.

Your setting **dateform =** .....

DELETE (DEL)

DELETE file(s)

Description Delete a file in the machine's data memory.

No warning is given, although files without the 'R' (Retrieved) attribute are protected against deletion. To delete a file without retrieving it first, use the CHMOD command to change the file attribute to 'R' (Retrieved).

Command syntax **delete =** *Filename.Ext* / **all**

Parameters *Filename.Ext* is a valid MS-DOS/GRPS filename (up to 8 characters) followed by a period (.) and an extension of up to three characters. The specified file must exist in the machine's data memory.

all will delete all files that are not currently Un-retrieved (U).

Example 1 **delete aug3191.v01**

Example 2 **delete all**

Keypad menu `Output Data / File Delete`

Related commands DELETE is used to delete files after they have been retrieved from the machine's memory, and for safety will only operate on files marked as Retrieved (R).

DIR lists all files and their attributes.

DOWNLOAD (or RETRIEVE) is used to retrieve files from the machine's memory, and when completed will mark each file as Retrieved (R).

Your notes

DETOFF

DETector cards OFF

Description	Switches off all detector cards, reducing power consumption, and clears any pending detect signals.
Essential when	when using gas detector cards (which are by default powered-on) and it is necessary to reduce power consumption.

Command syntax	**detoff** The command prompt changes from `B>`, `V>`, `I>` or `D>` to `Q>`.
Note	If a gas detector card has been turned off, it will need to be powered again for at least 24 hours before accurate results can be obtained.
Keypad menu	Not available
Related commands	DETON switches all detector cards on (see note above).

Your notes

DETON

DETector cards ON

Description	Switch on all detector cards to the default settings, clears any pending detect signals, and re-tunes detectors.
Essential when	when resetting all detector cards to a known configuration, or to reverse the action of DETOFF.

Command syntax	**deton** The command prompt changes from `Q>` to `B>`, `V>`, `I>` or `D>` (unless SENSORS is set to **none**).
Note	If a gas detector card has been turned off, it will need to be powered again for at least 24 hours before accurate results can be obtained.
Keypad menu	Not available
Related commands	DETOFF switches all detector cards off.

Your notes

DIR

File DIRectory

Description Displays a list of the files in the machine's memory.

Display syntax **dir**

Example response

```
Filename       Bytes     First Created    Last Changed     Attr
A1234B  .I01   12,770    06:21 01/01/91   13:21 01/01/91   R
A156X   .I01   34,133    13:31 01/01/91   07:03 06/02/91   U
DAY23AM .I03   1,297     07:03 06/02/91   12:53 07/03/91   O
3 File(s)      47,236    HH:MM DD/MM/YY  980,329 Bytes Free
```

Notes

Each filename has an extension, which begins with 'I' for interval files or 'V' for VBV files. This is followed by two numeric digits assigned by the machine in the range from 00 to 99.

Each file has an attribute indicating its current status:

R	Retrieved and closed
U	Un-retrieved and closed
O	Open and Un-retrieved

A closed file cannot be re-opened. An open file is closed by a PRINT, RETRIEVE or STOPREC command, or by the actions of BREAK, INTONOFF or VBVONOFF.

Use the CHMOD command to change the attribute of closed files to Retrieved or Un-retrieved.

The last line of the machine's response gives the file totals and free memory remaining, and identifies the current DATEFORM format.

Keypad menu	Not available
Related commands	CHMOD changes file attributes to ‘R’ or ‘U’.
	DOWNLOAD (RETRIEVE) is used to retrieve files from the machine’s memory, and when completed will mark each file as Retrieved (R).
	PRINT is used to display a file in the machine’s memory, and when completed will mark the file as Retrieved (R).
	All of the following commands will close an open file: DOWNLOAD (RETRIEVE), PRINT, STOPREC, BREAK, INTONOFF or VBVONOFF.

Your notes

DOWNLOAD (RETRIEVE)

DOWNLOAD or RETRIEVE file(s)

Description Retrieve one or more data files from the machine's memory.

The commands DOWNLOAD and RETRIEVE are alternative names with exactly the same effect.

Essential when memory will shortly become full, and before any further data processing can be done.

Command syntax **download** [*Filename.Ext* / **all** / **new** / **u** / **r** / **ur**]

Parameters

Filename.Ext is a valid MS-DOS/GRPS filename (up to 8 characters) followed by a period (.) and an extension of up to three characters. If the period and extension are omitted, all files with that *Filename* will be retrieved.

all will download all files in the machine's memory. Any files that are open will be closed, and a new file started with an incremented extension number.

new will download all files that have not been downloaded before (or not successfully) and are thus are currently marked Un-retrieved (U).

u will download all Un-retrieved files that are not currently open.

r will download all Retrieved files that are not currently open.

ur will download all Un-retrieved and Retrieved files that are not currently open.

When used with no parameter, **download** is equivalent to **download new**.

Notes

When using **download u** or **download ur**, remember that any files that are currently open will not be downloaded. Take care not to miss any files because of this.

A file download can be started at any time, but may be slower while the machine is actively recording vehicle data. If the download involves closing any currently open file(s), there will be a brief pause while corresponding new file(s) are opened with an incremented number in the filename extension. After this, further recordings can be made in the newly opened files while file transfer is taking place. However, no further keyboard/keypad commands can be entered until the retrieval process is complete.

The file transfer protocol is set with the PROTOCOL command (except that plain ASCII transfers are done with the PRINT command). After entering the DOWNLOAD or RETRIEVE command you must also to start the same file transfer protocol at the receiving device.

Files are transferred one at a time, Un-retrieved files first, followed by previously retrieved files, in both cases in order of date opened.

When each file transfer finishes successfully, the attribute of the retrieved file in the machine will be changed to ‘R’ (unless retrieved via the remote serial interface in **Read-only** mode – see INTERFACE).

Example 1 **download a34.v01**
Explanation Retrieves that one file

Example 2 **download a34-ps**
Explanation Retrieves all files which match the main filename, e.g. A34-PS.V01 and A34-PS.I02

Keypad menu Not available

Related commands RETRIEVE is the preferred name for this command, with exactly the same effect.

BREAK allows the same files to be easily recovered more than once, without interference between users.

CHMOD changes file attributes to ‘R’ or ‘U’.

DELETE is used to delete files after they have been retrieved from the machine’s memory, and for safety will only operate on files marked as Retrieved (R).

DIR lists all files and their attributes.

INTERFACE sets the security protection mode for the remote serial interface, and affects whether files are marked ‘Retrieved’ after a successful download.

Your notes

EOFCHARS

End Of File CHARacterS

Description	When an ASCII file is output using the PRINT command, some computer systems require a special sequence of characters to mark the end of the file. EOFCHARS specifies this sequence, if needed.
Essential when	the receiving computer system requires an end-of-file marker.

Command syntax	**eofchars =** [*Char1*] [*Char2*...]
Parameters	*Char1* etc. are ASCII characters, in decimal representation. A space is required between characters. If no characters are given (press Enter immediately after the '=') any previous end-of-file characters are deleted.
Valid range	Up to 12 ASCII decimal codes 00 to 127 (see Appendix 3)
Default	eofchars = 00 ASCII 'null', equivalent to no functional end-of-file characters
Example	**eofchars = 13 10**
Display syntax	**eofchars**
Example response	`EOFCHARS = 27 88`
Keypad menu	`General Setup / EOF Chars`
Related commands	EOLCHARS sets or displays the end-of-line characters. EOPCHARS sets or displays the page length and the end-of-page characters. PRINT generates an ASCII output file containing the above.

Your setting **eofchars =** .....

Your notes

EOLCHARS

End Of Line CHARacterS

Description	When an ASCII file is output using the PRINT command, most computer systems require a special sequence of characters to mark the end of each line. EOLCHARS specifies this sequence.
Essential when	the receiving computer system requires an end-of-line marker.

Command syntax	**eolchars =** [*Char1*] [*Char2*...]
Parameters	*Char1* etc. are ASCII characters, in decimal representation. A space is required between characters. If no characters are given (press Enter immediately after the '=') any previous end-of-line characters are deleted.
Valid range	Up to 12 ASCII decimal codes 00 to 127 (see Appendix 3)
Default	eolchars = 13 10 CR-LF, as required by MS-DOS
Example	**eolchars = 13**
Display syntax	**eolchars**
Example response	`EOLCHARS = 13 10`
Keypad menu	`General Setup / EOL Chars`
Related commands	EOFCHARS sets or displays the end-of-file characters. EOPCHARS sets or displays the page length and the end-of-page characters. OSP, OSP+ and PRINT generate ASCII output files containing the above.

Your setting **eolchars =** .....

Your notes

EOPCHARS

End Of Page CHARacterS

Description	The PRINT command requires information on the number of lines per page of output, and also any special characters required to start a new page. EOPCHARS specifies or displays the page length and the end-of-page characters.
Essential when	using the PRINT command to produce paged output.

Command syntax	**eopchars =** *PageLength* [*Char1*] [*Char2*...]
Parameters	*PageLength* is the number of lines per page. If *PageLength* is zero, output is continuous with no pagination. *Char1* etc. are ASCII characters, in decimal representation. A space is required between characters. If no characters are given after *PageLength*, any previous end-of-page characters are deleted.
Valid range	*PageLength*: 0 to 127 *Char1* etc.: up to 12 ASCII decimal codes 00 to 127 (see Appendix 3)
Default	eopchars = 60 12
Example	**eopchars = 66 27 54**
Display syntax	**eopchars**
Example response	`EOPCHARS = 84 27 88`
Keypad menu	`General Setup / EOP Chars`

Related commands EOFCHARS sets or displays the end-of-file characters.

EOLCHARS sets or displays the end-of-line characters.

PRINT generates an ASCII output file containing the above.

Your setting **eopchars =** .....

Your notes

EOVDACC

End Of Vehicle Detection by ACCeleration

Description

In tube or piezo sensor layouts where no loop is present to detect the front and rear of the vehicle, a number of parameters are needed to identify each separate vehicle from its axle-crossing data.

The EOVDACC command sets or displays the end-of-vehicle Acceleration parameter. EOVDACC is complementary to the end-of-vehicle Time-gap and Separation parameters EOVDGAP and EOVDSEP, and all three should be used together.

EOVDACC, EOVDGAP and EOVDSEP are only valid for axles-only sensor configurations; they are not relevant in other layouts.

Essential when

SENSORS specifies TT, TT*2, PP or PP*2.

Command syntax **eovdacc =** *SpeedChange*

Parameters

SpeedChange is the maximum absolute change in axle speed for the same vehicle (either acceleration or deceleration, in the current UNITS). If the computed speed change is any greater, the vehicle processor will assume that the newly-detected axle belongs to a different vehicle, and that the previous axle detection marked the end of the previous vehicle.

Valid range

1 to 99 km/hr or 1 to 62 mph

Default

eovdacc = 3 km/hr or 2 mph

Example

eovdacc = 10

Display syntax **eovdacc**

Example response

```
E0VDACC = 8
```

Keypad menu	`Tech Mode / EOVD Acc`
Related commands	EOVDGAP and EOVDSEP should accompany EOVDACC. EOVDSPEED is used for a separate purpose, to identify crossing or overtaking vehicles. UNITS specifies speed units of either kilometres/hour or miles/hour.

Your setting **eovdacc =** .....

Your notes

EOVDGAP

End Of Vehicle Detection by time GAP

Description

In tube or piezo sensor layouts where no loop is present to detect the front and rear of the vehicle, a number of parameters are needed to identify each separate vehicle from its axle-crossing data.

The EOVDGAP command sets or displays the end-of-vehicle Time-gap parameter. EOVDGAP is complementary to the end-of-vehicle Acceleration and Separation parameters EOVDACC and EOVDSEP, and all three should be used together.

EOVDACC, EOVDGAP and EOVDSEP are only valid for axles-only sensor configurations; they are not relevant in other layouts.

Essential when

SENSORS specifies TT, TT*2, PP or PP*2.

Command syntax **eovdgap =** *Gap*

Parameters

Gap is the maximum time in milliseconds within which the next tube detection must occur in order to be associated with the same vehicle. If the gap is any greater, the vehicle processor will assume that the newly-detected axle belongs to a different vehicle, and that the previous axle detection marked the end of the previous vehicle.

Valid range 100 to 20000 milliseconds

Default eovdgap = 1000 (i.e. 1 second)

Example **eovdgap = 500**

Display syntax **eovdgap**

Example response

```
EOVDGAP = 1500
```

Keypad menu	`Tech Mode / EOVD Gap`
Related commands	EOVDACC and EOVDSEP should accompany EOVDGAP. EOVDSPEED is used for a separate purpose, to identify crossing or overtaking vehicles.

Your setting **eovdgap =** .....

Your notes

EOVDSEP

End Of Vehicle Detection by axle SEParation

Description

In tube or piezo sensor layouts where no loop is present to detect the front and rear of the vehicle, a number of parameters are needed to identify each separate vehicle from its axle-crossing data.

The EOVDSEP command sets or displays the end-of-vehicle Separation parameter. EOVDSEP is complementary to the end-of-vehicle Acceleration and Time-gap parameters EOVDACC and EOVDGAP, and all three should be used together.

EOVDACC, EOVDGAP and EOVDSEP are only valid for axles-only sensor configurations; they are not relevant in other layouts.

Availability Version 1.9 onwards.

Essential when SENSORS specifies TT, TT*2, PP or PP*2.

Command syntax **eovdsep =** *Separation*

Parameters *Separation* is the maximum computed axle separation (in the current UNITS) that will be associated with the same vehicle. If the computed separation is any greater, the vehicle processor will assume that the newly-detected axle belongs to a different vehicle, and that the previous axle detection marked the end of the previous vehicle.

Valid range 100 to 2000 centimetres, or 39 to 787 inches

Default eovdsep = 700 centimetres or 276 inches

Example **eovdsep = 500**

Display syntax **eovdsep**

Example response `EOVDSEP = 400`

Keypad menu	`Tech Mode / EOVD Separation`
Related commands	EOVDACC and EOVDGAP should accompany EOVDSEP. EOVDSPEED is used for a separate purpose, to identify crossing or overtaking vehicles. UNITS specifies length units of either centimetres or inches.

Your setting **eovdsep =** .....

Your notes

EOVDSPEED

End Of Vehicle Detection by SPEED

Description	In tube or piezo sensor layouts where no loop is present to detect the end of the vehicle, and the sensor extends across more than one lane, EOVDSPEED is used to identify vehicles that are crossing or overtaking at the sensor position. EOVDSPEED is only valid for axles-only sensor configurations; it is not relevant in other layouts.
Availability	Version 1.9 onwards.
Essential when	SENSORS specifies TT, TT*2, PP or PP*2.

Command syntax	**eovdspeed =** *Speed*
Parameters	*Speed* is the maximum computed speed (in the current UNITS) that could result from the next axle detection. If the apparent speed exceeds the EOVDSPEED setting, the vehicle processor will assume there are two crossing or overtaking vehicles.
Valid range	80 to 320 kilometres/hour or 50 to 199 miles/hour
Default	eovdspeed = 240 kilometres/hour or 150 miles/hour
Example	**eovdspeed = 185**
Display syntax	**eovdspeed**
Example response	`EOVDSPEED = 180`

Keypad menu `Tech Mode / EOVD Speed`

Related commands EOVDACC, EOVDGAP and EOVDSEP are used together in axles-only sensor configurations to distinguish between following vehicles.

UNITS specifies speed units of either kilometres/hour or miles/hour

---

Your setting **eovdspeed =** .....

Your notes

FILENAME

Description All recordings share a common filename, and the number in the filename extension is incremented automatically by the machine. The FILENAME command sets and displays the filename being used.

Essential when you wish to use a filename other than the site/date name generated by the machine.

Command syntax **filename =** [*Filename*]

Parameters *Filename* is a valid MS-DOS/GRPS filename (up to 8 characters).

If no filename is specified, the machine will generate its own, based on the SITE name and date. The format will be *SSSYMMDD*: *SSS* are the first three characters of the SITE name; *Y* is the last digit of the year; and *MM* and *DD* are the month and day the file was opened.

Notes FILENAME can also be changed by the STARTREC command.

Valid range Valid characters for *Filename* include any combination of the following: 0 to 9, A to Z, a to z and _ (underline)

Default Automatic site/date filename

Example 1 **filename = coldlane**

Example 2 **filename =**
Explanation Revert to automatic site/date filenames

Display syntax **filename**

Example response `FILENAME = FILE1234`

Keypad menu	`Start Survey / Filename`
Related commands	SITE specifies the site name that will be used in automatic file names. STARTREC will start a recording and can optionally change FILENAME at the same time.

Your setting **filename =** .....

Your notes

GAS

CO GAS sensor calibration

Description The GAS command changes the calibration of the CO sensor. It is used twice: once to zero the sensor, and again to enter the CO level when the sensor is filled with a test gas of known composition.

Availability Version 1.9 onwards.

Essential when calibrating the CO sensor. Calibrate the zero setting before every survey, and calibrate to a known gas concentration at least every 6 months.

Command syntax **gas = 0**

gas co1 = *COlevel*

Parameters **0** is entered when the CO sensor is filled with a known CO-free gas – i.e. **not** the air at the monitoring site.

COlevel is the CO concentration in parts per million of the test gas filling the sensor.

Valid range *COlevel* can range from 20 to 35 parts per million

Examples gas = 0
gas co1 = 25

Response In both cases the initial response is:
`Calibrating Sensor Please Wait....`
Calibration can take up to 5 minutes, and is normally followed by:
`Calibration complete`

Keypad menu Not available

Related commands GASCAL sets and displays the calibration factors for the CO and temperature sensors.

MONITOR GAS provides real-time indications, peak and running-average values of temperature and CO levels.

Your notes

GASCAL

GAS sensor CALibrations

Description GASCAL sets and displays the calibration factors for the CO and temperature sensors.

After every calibration, make a note of the calibration factors reported by GASCAL. If these values ever need to be re-entered, e.g. after a complete loss of power, use GASCAL to re-enter them.

Availability Version 1.9 onwards.

Essential when previous CO calibration values have become lost or corrupted.

Command syntax **gascal co1 =** *TempComp COGradient ZeroTemp ElectrZero*

Parameters **co1** is required as part of the command.

TempComp is the compensation factor for the temperature sensor. It is factory set and should be re-entered as before.

COGradient is the sensitivity of the CO detector, which is established by the calibration routine using the GAS command. *COGradient* can then be read and noted, and re-entered here after a power or memory failure.

ZeroTemp is the zero setting for the temperature sensor. It is factory set and should be re-entered as before.

ElectrZero is the electrical zero of the CO detector, which is established by the calibration routine using the GAS command. *ElectrZero* can then be read and noted, and re-entered here after a power or memory failure.

Defaults Factory-set for each machine

Example **gascal co1** = 21 1200 0 115

Display syntax	**gascal co1**
Example response	`GASCAL = 20 1000 0 100`
Keypad menu	Not available.
Related commands	GAS is used to calibrate the CO sensor with a test gas mixture. MONITOR GAS provides real-time indications, peak and running-average values of temperature and CO levels.

Your settings

Temperature sensor compensation factor (*TempComp*)

.....

Sensitivity of CO detector (*COGradient*)

.....

Temperature sensor zero (*ZeroTemp*)

.....

Electrical zero of CO detector (*ElectrZero*)

.....

Your notes

GRIDREF

GRID REFerence

Description

GRIDREF sets or displays the optional site Grid Reference. Although intended primarily for use with the UK National Grid, any other alphanumeric information can be inserted.

GRIDREF will be used to generate mapping data in future versions of the *Showman Plus* analysis software.

Command syntax **gridref =** [*Text*]

Parameters *Text* can be any alphanumeric characters, punctuation marks or spaces. All lower-case letters are converted and stored as upper-case.

Valid range Up to 19 characters; any excess will be truncated with `Warning 01 : String has been truncated`.

Example **gridref = SU234123**

Display syntax **gridref**

Example response `GRIDREF = NY430140`

Keypad menu `Start Survey / Grid Reference`

Your setting **gridref =** .....

Your notes

HEADINGS

Lane HEADING descriptionS

Description HEADINGS will set or display the optional descriptions for the directional heading or position of each lane.

Command syntax **headings =** [*Lane1*] [*Lane2...*]

Parameters *Lane1* is an alphanumeric code. Examples are symbols such as ‘NW’, compass bearings divided by 10 (e.g. ‘29’ for 290°) or SHRP-type lane descriptions such as ‘A50’, ‘D51’.

Lane2 etc. are similar codes for each successive lane in order. Each code must be separated from the next by a space.

If no headings are supplied (press **Enter** immediately after ‘=‘) any previous HEADINGS information is deleted.

Valid range A maximum of 8 codes, up to 4 characters per lane

Example 1 **headings = 0 0 0 18 18 18**
Explanation Six lanes

Example 2 **headings =**
Explanation Clears previous settings

Display syntax **headings**

Example response `HEADINGS = NW NW NW SE SE SE N N`

Keypad menu `Start Survey / Headings`

Your setting **headings =** .....

Your notes

HELP (ALL)

Description

HELP lists all the commands in alphabetical order. If a command has variable parameters, the settings are displayed. For commands without variables, a brief line of text beginning with REM explains the function.

HELP is a useful reminder of the format of each command.

To terminate the output before the end, press **Esc** or **Ctrl-C**.

Display syntax **help**

Example response

HELP does not display any parameters related to sensor cards that are not fitted to the machine. For example, the GAS and GASCAL commands will only be displayed if the machine contains a Pollution Monitor card.

```
ACTBINS = 0 2 3 4 5 6 7 8 99
REM ALL,HELP - Displays this information
AXFACTOR = 2.000
AXFILTER = 20
AXSEPS = 400 400 400 400 400 400 400 400
BREAK = DAILY
REM CFGLOAD - Loads a stored configuration
REM CFGSAVE,CFGSTORE - Stores a configuration
CHANNELS = 1 1 1 1 1 1 1 1
```

etc.

etc.

```
WTMODE = 1
WTOVERS = 2
WTTHRES = 8
WTUNDERS = 2
WTWINDOW = 16
```

You are then returned to the current GRPS input prompt.

Keypad menu Not available

Related commands ALL is an alternative command name for HELP.

PAGELEN sets the number of lines of text, after which the output pauses with a `------ More ------` prompt.

REM is a comment line which is ignored.

STATUS lists the machine's hardware status.

INTERFACE

Description

INTERFACE sets the level of security protection for access via the serial communications ports (COM1 and COM2). This may be desirable when multiple users need to retrieve data from a remote machine, to protect against accidental or unauthorised deletion of data or reprogramming of the machine.

The machine can be set into one of three modes:

Read-write Full access to all commands and functions. When files are retrieved via the modem link in this mode, their status is changed from 'Un-retrieved' to 'Retrieved'.

Read-only The user can retrieve files and display the machine status but cannot alter any settings. Any files retrieved in this mode will remain marked as 'Un-retrieved', so that further users can use commands such as **retrieve u** to retrieve the same files.

Locked The serial port gives access to no functions except INTERFACE to change the protection mode.

Changing the mode to either **Read-write** or **Read-only** requires a password.

Protection set by INTERFACE has no effect on local access via the keypad. If the machine is restarted using the keypad menu (`Esc-Edit-0`), or using the serial port hard restart switch, it resets the interface to **Read-write** and removes password protection.

Availability Version 1.9 onwards.

Essential when security against accidental or unauthorised remote commands is required.

Command syntax **interface = rw** [*RWpassword*] /
ro [*ROpassword*] /
lock

Parameters

rw sets **Read-write** mode, with full access to all functions

RWpassword is the password required to change the machine into **Read-only** mode. *RWpassword* is itself set using the PASSWORD command.

ro sets **Read-only** mode, with access to retrieve files and read all machine settings via the remote serial port, but no access to change any settings (except **interface rw** using *RWpassword*).

ROpassword is the password required to change the machine into **Read-only** mode. *ROpassword* is itself set using the PASSWORD command.

lock sets the machine into **Locked** mode. The GRPS command prompt is L>, and **Locked** mode gives no access via the remote serial port to any function except **interface ro** or **interface rw** (requiring the appropriate password).

Default

interface = rw

Notes

Passwords are case-sensitive, so ‘PASSWORD’, ‘password’, ‘Password’ and ‘PassWord’ are all different.

If you use an incorrect password with the INTERFACE command, there will be an error message:

```
Error 70: Incorrect password
```

If you attempt to use any unauthorised command in **Read-only** mode, the error message will be:

```
Error 71: Command not accessible in this mode
```

No error messages are displayed in **Locked** mode.

Example

interface = rw Daylight

Explanation

‘Daylight’ (with only a capital D) must be the correct password for changing to **Read-write** mode.

Display syntax **interface**

Example response

```
INTERFACE = RO
```

Keypad menu Not available; keypad control is not password-protected.

Related commands PASSWORD sets the two separate passwords to change to **Read-write** mode or to **Read-only** mode.

TIMEOUT can set a time delay after which the remote serial interface will automatically revert to **Locked** mode if nothing has been received at the input.

Your setting **interface =** .....

Your notes

INTERVAL

Description A recording interval is the time period over which data are collected before being 'binned' together as a single total. INTERVAL sets or displays this period and ensures that it is synchronized, both with the time of day starting at 00:00 and with the period between file breaks if BREAK has been set.

Typical INTERVAL values are 5, 15 or 60 minutes.

Essential when setting up for recording.

Command syntax **interval =** *IntervalTime*

Parameters *IntervalTime* is the length of the recording interval in minutes.

Valid range *IntervalTime* must be greater than or equal to the Peak Interval (PEAKINT), regardless of whether peak period recording is actually being used. If the interval between peak-period recordings is to be shorter, it must be an exact sub-multiple of *IntervalTime*.

IntervalTime must be less than or equal to the time between file breaks (set using BREAK). If less than the BREAK time, *IntervalTime* must be an exact sub-multiple.

Within these limits, the valid range for *IntervalTime* is 1 to 1440 minutes.

Default interval = 15

Example 1 **interval = 10**
Explanation Compatible with PEAKINT times of 1,2, 5, 10 minutes and BREAK times of 10, 20, 30... minutes

Display syntax **interval**

Example response `INTERVAL = 20`

Keypad menu `Start Survey / Interval`

Related commands BREAK sets the time between automatic file breaks.

PEAKINT sets the recording interval during peak periods, which is usually shorter than INTERVAL.

PEAKTIME controls when peak-period recording begins and ends.

Your setting **interval =** .....

Your notes

INTFILTER

INTerval data FILTER

Description GRPS allows one logical criterion to be applied when selecting vehicle data to be recorded in interval (INT) data files. INTFILTER specifies this criterion, or displays the current setting.

Essential when selective recording is required in interval files.

Syntax **intfilter = all** / *Condition*

Parameters

all disables interval data filtering and records all vehicles.

Condition is a single logical condition that a vehicle must meet in order to be recorded. This format of this condition is:

keyword comparator value

The valid keywords are:

ACT	Axle count
ASP	Axle separation (largest value for the vehicle)
AWT	Axle weight (largest value for the vehicle)
CLS	Classification
DRN	Direction of travel (1 = normal direction, 2 = reverse direction)
GWT	Gross weight
LAN	Lane number
LEN	Overall length
SPD	Speed
WBT	Overall wheelbase

The valid comparators for use with keywords are:

>	is greater than (but is not equal to)
>=	is greater than, or is equal to

	=	is equal to (use only with ACT, CLS, DRN or LAN)
	<=	is less than, or is equal to
	<	is less than (but is not equal to)
	<>	is not equal to (use only with ACT, CLS, DRN or LAN)

See the examples below for typical applications.

Valid range The *Condition* must be compatible with the UNITS setting and the sensors available.

Default intfilter = all

Example 1 **intfilter = spd > 120**
Explanation Record only vehicles with speed greater than 120, in the current UNITS of speed

Example 2 **intfilter = cls = 2**
Explanation Record only vehicles in class 2

Example 3 **intfilter = cls <> 2**
Explanation Ignore vehicles in class 2

Display syntax **intfilter**

Example response `INTFILTER = ACT < 3`
Explanation Record only vehicles with 2 axles

Keypad menu `Start Survey / INT Filter`

Related commands OSPFILTER, TTEFILTER and VBVFILTER.

Your setting **intfilter =** .....

INTONOFF

INTerval recording ON and OFF

Description INTONOFF controls the dates and times when interval recordings will start and finish.

Essential when programming an instrument to be left to record data at some later time.

Command syntax **intonoff =** *StartTime StartDate EndTime EndDate* / **off**

Parameters

StartTime is the time at which recording is to start, in the 24-hour format 'hh:mm'.

StartDate is the date on which recording is to start, in the format specified by DATEFORM.

EndTime is the time at which recording is to end, in the 24-hour format 'hh:mm'.

EndDate is the date on which recording is to end, in the format specified by DATEFORM.

off cancels any previous settings.

intonoff is automatically set to **off** when the programmed recording is complete (i.e. *Endtime-EndDate* has passed).

Valid range 00:00 01/01/96 (1996) to 23:59 31/12/95 (2095), or equivalent according to the DATEFORM format.

StartTime/StartDate must be before *Endtime/EndDate*.

An input of 24:00 will be interpreted and stored as 00:00 of the next day. Starting with *Marksman* firmware version 1.94, two-digit year numbers must be 96 or later, in order to be interpreted correctly (as 1996 to 2095). The year can also be entered in four-digit format.

Default intonoff = off

➲ CAUTION

After setting INTONOFF, you must then enter STARTREC INT to enable interval recording. This will open a file but no data will be entered until StartTime/StartDate is reached.

Example **intonoff =** 00:00 01/04/98 24:00 30/04/98
Explanation Recording will stop at 00:00:00 on 01/05/98.

Display syntax **intonoff**

Example response `INTONOFF = 00:00 01/04/98 00:00 01/05/98`

Keypad menu `Start Survey /  INT On/Off`

Related commands DATEFORM sets the format for all dates used in GRPS.

STARTREC enables recording from the time set by INTONOFF.

STOPREC stops or prevents all recording, regardless of the INTONOFF setting. STOPREC INT is specific to interval recording.

Your setting **intonoff =** .....

Your notes

INTSPEC

INTerval recording SPECification

Description When more than one type of data is available from the sensor layout, INTSPEC allows interval recording data to be collected in various combinations for later analysis.

Only certain sensor configurations are capable of providing information in the form required. The table on page 119 shows almost every practical sensor configuration; in each case, INTSPEC should only call for those data items marked with a ●.

Essential when more than one type of data is available from the sensor layout.

Command syntax **intspec =** *Parameter* [★ / **+** *Parameter2*]
[★ / **+** *Parameter3*]

Parameters *Parameter* is one of the list below. *Parameter2* is a different selection from the list, and so on.

ACT	Axle count
AWT	Axle weight (largest value for the vehicle)
CLS	Classification
CNT	Count
CO1	CO level (combine using '+' only)
DRN	Direction
GAP	Gap (in time) from front of this vehicle to rear of previous vehicle in the same lane
GWT	Gross weight
HWY	Headway (in time) from front of this vehicle to front of previous vehicle in the same lane
LEN	Overall length (in current UNITS)
SDS	Speed counts with statistics (mean speed and

		standard deviation)
	SPD	Speed (in current UNITS)
	WBT	Overall wheelbase (in current UNITS)

Combinations

Parameters may be combined using either '★' or '+'.

A★*B* means that two parameters *A* and *B* will be stored together for each vehicle, in a single two-dimensional table. During analysis, both *A* and *B* can be identified for each separate vehicle.

A+*B* means that parameter *A* will be added to one total for all vehicles counted, and parameter *B* added to a different total. This is much more economical in storage (only the two totals are stored) but during analysis there is no way of re-associating *A* and *B* with any individual vehicle.

'★' takes precedence over '+' (see further examples below).

Valid range

No more than three parameters can be combined. Selections that are not possible with the sensor cards fitted to the machine will produce an error message. The table on page 119 shows the available combinations for almost all practical sensor layouts.

CAUTION

There are additional limitations when using the Showman Plus for Windows *data analysis software – see below.*

Default

intspec = cnt

Example 1

intspec = cls + spd

Example 2

intspec = cls ★ gap + cnt

Explanation

Two-dimensional table of CLS and GAP, and a separate one-dimensional table for CNT.

Display syntax

intspec

Example response

```
INTSPEC = CLS * SPD + HWY
```

Memory usage

Each parameter measured (except a simple count) will usually be stored in one of several 'bins' according to its value. Memory usage depends on the total number of bins required, and the way that parameters are combined.

The following examples assume that each parameter uses 10 bins.

CNT	Table of Vehicle Counts only (total 1 bin)
CLS + SPD	Separate tables of Class and of Speed (total 10 + 10 = 20 bins)
CLS ★ SPD	Speed for each vehicle (total 10x10 = 100 bins)
CLS ★ GAP + CNT	Two-dimensional table holding both Class and Gap, and a separate Count table (total 10x10 + 1 = 101 bins)

➲ CAUTION

If you intend to import the recorded data into Golden River's Showman Plus for Windows *data analysis software, there is a limit of 255 bins. Consult the* ***Showman Plus for Windows User Manual*** *for details.*

Keypad menu

`Start Survey / INT Spec`

Related commands

ACTBINS, LENBINS, SPDBINS, TIMEBINS and WTBINS specify the numbers of bins for their respective measurements, and interact with INTSPEC to determine memory usage.

Your setting **intspec =** .....

Sensor Configurations

	T **T*2** **P** **p** **P*2**	**TTN** **TTN*2**	**L** **L*2** **N+1*2** **N+1*3** **N+1*4**	**LL** **2(N+1)*2** **2(N+1)*3** **2(N+1)*4**	**TT** **TT*2** **pp** **PP** **PP*2**	**TLT** **LTL** **pLp** **PLP** **LpL**	**aa**	**L2aL** **2aL2a** **L2aL2** **LaLa** **LaL**	**ww**	**LwL** **L2wL** **LwLw** **L2wL2w**
ACT Axle Count					●	●	●	●	●	●
AWT Axle Weight									●	●
CLS Class				EUR6 only	●	●	●	●	●	●
CNT Count	●	●	●	●	●	●	●	●	●	●
CO1	Carbon monoxide measurement is available if a gas sensor card is fitted									
DRN Direction		●		●	●	●	●	●	●	●
GAP Gap			●	●	●	●	●	●	●	●
GWT Gross Weight									●	●
HWY Headway			●	●	●	●	●	●	●	●
LEN Length				●		●		●		●
SDS Speed + statistics				●	●	●	●	●	●	●
SPD Speed				●	●	●	●	●	●	●
WBT Wheel-base Total					●	●	●	●	●	●

LANGUAGE

Description

GRPS is capable of accepting input and producing messages in more than one language.

LANGUAGE specifies which language is to be used. The examples given assume that the current language is English.

Command syntax **language =** *Language*

Parameters

Language can be one of the following:

english Available now

francais
deutsch
italiano
espanol
— Check availability with Golden River

LANGUAGE is never reset automatically to the default, even by a RESTART 4 command; any change must be entered manually.

Default language = english

Example **language = francais**

Display syntax **language**

Example response `LANGUAGE = English`

Keypad menu `General Setup / Language`

Your setting **language =** .....

Your notes

LENBINS

LENgth BINS

Description Vehicles whose length or wheelbase falls within a specified range can be grouped together in the same 'bin' for counting purposes. LENBINS specifies the length ranges.

After defining the bin boundaries using LENBINS, use the LEN parameter in the INTSPEC command to enable length and/or wheelbase classification.

Essential when INTSPEC includes LEN or WBT.

Command syntax **lenbins =** *LowestBin* [*BinBoundary*...] *HighestBin*

Parameters *LowestBin* is the shortest length or wheelbase that will be included in the lowest bin. 0 must be supplied if required.

BinBoundary is the shortest length or wheelbase that will be included in the bin **to the right**.

HighestBin is slightly higher than the longest length or wheelbase that will be included in any bin. (A vehicle with a measured length or wheelbase exactly equal to *HighestBin* will not be recorded.)

Valid range Up to 15 bin boundaries (defining up to 14 bins)

Length/wheelbase 0 to 9999 centimetres or 3936$ inches, according to the current setting of UNITS.

Bin boundaries must be entered in increasing order.

Default lenbins = 0 300 450 600 900 9999 centimetres

lenbins = 0 118 177 236 354 3937 inches

Example **lenbins = 0 180 9999**
Explanation Separates cycles and motorcycles from all longer vehicles (UNITS are metric).

Display syntax **lenbins**

Example response `LENBINS = 0 400 800 1500 9999`

Keypad menu `Start Survey / Bins, Length`
The display shows the units and the present number of bins.

Related commands INTSPEC is used to select this type of classification.

UNITS specifies the length units, centimetres or inches.

Your setting **lenbins =** .....

Your notes

LOCATION

Description

The Site Location description accompanies the data file and is used in the Showman Plus data analysis software. For example it can be printed in the title of a table or graph.

LOCATION edits this optional text.

Command syntax **location =** [*LocationText*]

Parameters

LocationText may contain up to 19 alphanumeric characters, including spaces or punctuation symbols. Alphabetic text will be converted to UPPER CASE. Input longer than 19 characters will be truncated with a `Warning 01 : String has been truncated` message.

If no *LocationText* is supplied (press Enter immediately after the '=') the previous text will be deleted.

Example **location = rough hill road**
Explanation Stored as ROUGH HILL ROAD

Display syntax **location**

Example response

```
LOCATION = SUNRISE HIGHWAY
```

Keypad menu `Start Survey / Site Location`

Related commands GRIDREF, HEADINGS and SITE provide other text descriptions of the site.

Your setting **location =** .....

Your notes

LP2THRES (CHHIGH%)

LooP, class 2 THREShold

Description	LP2THRES is a sensor 'tuning' command, for use with loop-loop classification schemes only (e.g. EUR6). LP2THRES sets or displays the threshold value between detection signals that are counted as a 'low chassis' and those counted a 'high chassis'. This setting must be adjusted to suit the site and typical vehicle types. To assist in this, LPCLS2 displays the running-average signal levels for the last 100 cars, and CHAUTCNT (CLS2CNT) verifies that sufficient vehicles have been counted.
Availability	Version 1.9 onwards. The preferred name CHHIGH% was introduced in version 1.98.
Essential when	calibrating a loop-loop sensor system for classification.

Command syntax	**lp2thres =** *Threshold* / [*Threshold...*]
Parameters	*Threshold* is the running-average change in LPCLS2 expressed as a percentage. A single value applies to all lanes, but separate values can be set for up to eight lanes in sequence.
Note	If the signal from a particular vehicle is higher than (LPCLS2 x LP2THRES %), that vehicle will be counted as a 'low' chassis. Therefore if the system is counting some high-chassis vehicles as 'low-chassis', increase LP2THRES; if the system is counting some low-chassis vehicles as 'high-chassis', decrease LP2THRES.
Valid range	1 to 100
Default	lp2thres = 50 50 50 50 50 50 50 50
Example *Explanation*	**lp2thres = 45** Same value for all lanes (or else there is only one lane)
Display syntax	**lp2thres**

Example response	`LP2THRES = 65 63 50 50 50 50 50 50`
Explanation	Two lanes have been altered; the rest are still at the default value of 50
Keypad menu	Use `Tech Mode / CHHIGH%`
Related commands	CHHIGH% is now the preferred name for this command. CLASS selects the loop-loop classification scheme (e.g. EUR6). CHAUTCNT (CLS2CNT) and CHAUTAVLOW (LPCLS2) are the associated sensor monitoring commands.

Your setting **lp2thres =** .....

Your notes

LPCALC

LooP CALCulation

Description With loop detectors, vehicle length calculations are normally based on the time-tag of the ‘on’ event as the vehicle passes over the leading edge of loop. However, if the ‘off’ event at the trailing edge is more definite, LPCALC can make the vehicle processor use that time-tag instead. Alternatively LPCALC can be set to average the time-tags of events at both the leading and trailing edges of the loop.

These options are seldom required if using the loop detector cards in the marksman itself. They are provided mainly for use with external loop detectors whose outputs are connected to the *Marksman* through a switch card.

Availability Version 1.9 onwards.

Essential when external loop detectors are attached to a switch card (with **sensors = LL** and **swinput = loop**) and the normal leading-edge option is not satisfactory.

Command syntax **lpcalc = leading / trailing / average**

Parameters **leading** bases calculations on the time-tags of the ‘on’ events at the leading edge of the loop.

trailing bases calculations on the time-tags of the ‘off’ events at the trailing edge of the loop.

average bases calculations on the time-average of the ‘on’ and ‘off’ events.

Default lpcalc = leading

Note LPCALC can only be used when **sensors = LL**. With other loop sensor layouts, calculations are always based on the leading edge of the loop.

Example 1 **lpcalc = trailing**

Display syntax	**lpcalc**
Example response	`LPCALC = Average`
Keypad menu	`Tech Mode / LPCALC`
Related commands	SWINPUT selects loop or tube input from an external detector via a switch card.

Your setting **lpcalc =** .....

Your notes

LPCLS2 (CHAUTAVLOW)

LooP, CLaSs 2

Description	LPCLS2 is a display-only command for sensor 'tuning', for use with the loop-loop classification schemes only. LPCLS2 displays the running-average change in peak sensor output for the last 100 cars. To calibrate the chassis height sensing which is used to determine 'low' or 'high' chassis, the LPCLS2 value is compared against the threshold between 'high chassis' and 'low chassis' which is set using CHHIGH% (LP2THRES).
Availability	Version 1.9 onwards. Preferred name is now CHAUTAVLOW.
Essential when	calibrating a loop-loop sensor system for classification.

Display syntax	**lpcls2**
Example response	`LPCLS2 = 182 174 160 190 0 0 0 0`
Explanation	One value for each lane in use, up to eight lanes
Note	The running average result from LPCLS2 is not valid until CHAUTCNT (CLS2CNT) verifies that at least 100 cars have been counted.
Keypad menu	`Use Tech Mode / C/H Auto Avg Low`
Related commands	CHAUTAVLOW is now the preferred name for this command. CLASS selects the loop-loop classification scheme (e.g. EUR6). CHAUTCNT (CLS2CNT) verifies that at least 100 cars have been counted. CHHIGH% (LP2THRES) is the associated sensor threshold command.

Your notes

LPCYCLES (LPSENS)

Description	LPCYCLES governs the length of time for which each loop is scanned. The interval between scans of the same loop is a compromise: too long an interval will tend to miss fast-moving vehicles, while too-rapid scanning is likely to make the loops more susceptible to 'noise' interference and inaccurate detection
Availability	Version 1.9 onwards.
Essential when	setting very high sensitivity in conjunction with LPTHRES.

Command syntax	**lpcycles =** *Interval...[Interval]*
Parameters	*Interval* is essentially an arbitrary number, ranging from 1 (as slow as possible) to 255 (as fast as possible). If the machine contains a second loop detector card a different *Interval* can be set if required to optimise site setup. (If not set assumed to be same for both cards).
Valid range	1 to 255
Default	lpcycles = 208 208 (192 for loop boards manufactured before September 1993 – check using STATUS) [196 for 83 kHz 4 & 8 loop boards introduced for DBFO contracts]
Example	**lpcycles = 50** (one loop card or two loop cards set to the same *Interval*) **lpcycles = 50 60** (two loop cards set to different *Intervals.* For example if two 8 loop cards were fitted, loops 1-8 on connector 1 would be set to an *Interval* of 50 and loops 9-16 on connector 2 would be set to an *Interval* of 60).

Display syntax	**lpcycles**
Example response	`LPCYCLES = 128 128`
Keypad menu	`Tech Mode / LPCYCLES`
Related commands	LPSENS is an older name for LPCYCLES, but may be phased out because it is less descriptive. All the other 'LP-' commands can also be used for loop sensor calibration and 'tuning' if required.
Your setting	**lpcycles =** .....

Your notes

LPHOLD

LooP detector HOLD time

Description	A loop sensor detects metal, so if a vehicle remains static above the loop it will continue to give a signal. Until that signal falls again, the vehicle will not be counted. It is advisable that the detection circuit resets itself after a certain period of time; this is determined by LPHOLD. A typical value for LPHOLD would be 60 seconds, but for simple counting applications LPHOLD can be set much shorter. A minimum sensible time is 3-10 seconds.
Essential when	using loop detectors in locations where traffic is liable to be halted.

Command syntax	**lphold =** *Time*
Parameters	Time is the maximum period (in seconds) of continuous detection, after which the loop detector will reset itself.
Valid range	0 to 3000
Default	lphold = 180
Example	**lphold = 80**
Display syntax	**lphold**
Example response	`LPHOLD = 240`
Keypad menu	`Tech Mode / LPHOLD`
Related commands	All the other ‘LP-’ commands can also be used for loop sensor calibration and ‘tuning’ if required.

Your setting **lphold =** .....

Your notes

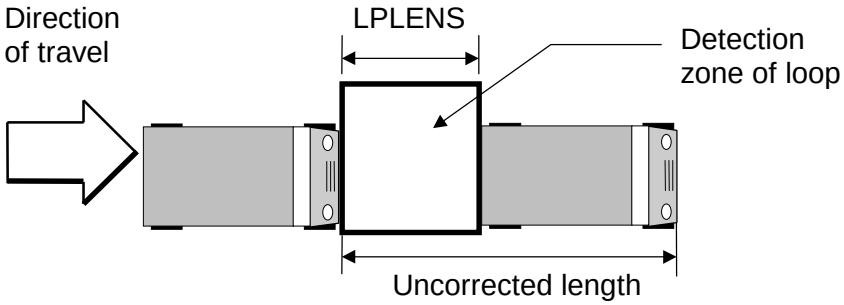
LPLENS

LooP detector LENgthS

Description

LPLENS notifies the machine of the length of the loop in each lane, as measured along the direction of travel.

Because the vehicle is detected from the moment the front crosses the leading edge of the loop, until the moment the rear leaves the trailing edge of the loop, the apparent vehicle length is its true length plus the length of the loop itself. LPLENS allows a correction to be made.



Essential when

using loop detectors and INTSPEC involves more than a simple vehicle count.

Command syntax

lplens = *Length1* [*Length2...*]

Parameters

Length1 is the size of the loop in lane 1 (closest to the machine) in the current UNITS of centimetres or inches, measured along the direction of travel.

If no further values are given, *Length1* is applied to all lanes.

If there is more than one loop in the same lane, enter the average length.

Length2 is the loop length in lane 2, and so on in sequence for all other lanes if applicable.

Valid range

10 to 999 centimetres, or 4 to 393 inches; up to 16 loops.

Default

lplens = 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 centimetres

lplens = 79 79 79 79 79 79 79 79 79 79 79 79 79 79 79 79
inches

Example **lplens = 201 198 204 200 201 200**

Display syntax **lplens**

Example response `LPLENS = 240 245 238 200 200 200 200 200`

Keypad menu `Start Survey / Loop Length`

Related commands All the other ‘LP-’ commands can also be used for loop sensor calibration and ‘tuning’ if required.

Your setting **lplens =** .....

Your notes

LPMODE

LooP detector MODE

Description The LPMODE command is only needed when two Loop cards are fitted to a *Marksman* unit. When two Loop cards are fitted, the cards scan the loops in either 'parallel' or 'serial' mode.

In parallel mode, both Loop boards scan their loops at the same time. In serial mode, each Loop board waits for the other one to finish scanning before it starts scanning its own loops.

Serial mode is only required if there is interaction between the loops attached to different Loop cards.

Essential when The *Marksman* unit has two Loop cards, and there is interaction between the loops attached to the different cards.

Command syntax **lpmode = 0 / 16**

Parameters

0 Loops are scanned in parallel mode

16 Loops are scanned in serial mode

Valid range 0–31, but values other than 0 and 16 may have undesired effects

Default lpmode = 0

Example **lpmode = 16**

Display syntax **lpmode**

Example response `LPMODE = 16`

Keypad menu `Tech Mode / LPMODE`

Related commands All the other 'LP-' commands can also be used for loop sensor calibration and 'tuning' if required.

Your setting **lpmode =** .....

Your notes

LPOVERS

Description

Loop detection is subject to electrical noise which may produce false ‘detect’ signals of short duration. Each loop is scanned repeatedly and vehicle detection is only considered genuine after a number of successive scans have given a positive response.

LPOVERS sets or displays the number of successive positive responses required.

Essential when

problems of noise or signal ‘dropout’ with loop detectors cannot be corrected using LPCYCLES and LPTHRES.

Command syntax **lpovers =** *Number* [*Number*]

Parameters

Number is the number of successive scans that must find a ‘detect’ condition in order for detection to be recorded.

If the machine contains a second loop detector card, a second, different, *Number* can be set if required to optimise site setup. (If not set assumed to be same for both cards).

Valid range

1 to 15

Note

Do not set to less than 2

Default

lpovers = 3 3

Examples

lpovers = 8

(one loop card or two loop cards set to the same *Number*)

lpovers = 9 12

(two loop cards set to different *Numbers.* For example if two 8 loop cards were fitted, loops 1-8 on connector 1 would be set to a *Number* of 9 and loops 9-16 on connector 2 would be set to a *Number* of 12).

Display syntax **lpovers**

Example response `LP0VERS = 9 12`

Keypad menu `Tech Mode / LP0VERS`

Related commands LPUNDERS should be set to the same value(s) as LPOVERS.

For loop boards manufactured since September 1993, LPUNDERS is automatically set to be the same as LPOVERS.

Your setting **lpovers =** .....

Your notes

LPSCAN

LooP detector SCANning interval

(Loop detector boards after September 1993 only – check using STATUS)

Description

LPSCAN sets or displays the time interval between successive scans of the same loop – the length of the scanning cycle. The detector card will scan each loop as quickly as possible and then, if time is available, will remain dormant until the end of the LPSCAN period.

The interval between scans of the same loop is a compromise: too long an interval will tend to miss fast-moving vehicles, while too-rapid scanning will consume unnecessary power and may prematurely drain the battery.

Availability

Version 1.9 onwards.

Essential when

count only sites where long battery life is important lpscan=30 to 40 typically.

speed and length classification and/or traffic speeds high lpscan=0

Command syntax **lpscan =** *Interval*

Parameters

Interval is the time in milliseconds between successive scans of the same loop. If there are too many loops in the layout to allow a complete scanning cycle within the time specified, scanning takes place as quickly as possible with no dormant period between scans.

Valid range

0 to 100 milliseconds

Default

lpscan = 0 (scan as rapidly as possible)

Example

lpscan = 30

Display syntax	**lpscan**
Example response	`LPSCAN = 10`
Keypad menu	`Tech Mode / LPSCAN`
Related commands	All the other ‘LP-’ commands can also be used for loop sensor calibration and ‘tuning’ if required.

Your setting **lpscan =** .....

Your notes

LPSENS (LPCYCLES)

Description	LPSENS has been renamed LPCYCLES. The machine will respond to LPSENS commands, but will give output in terms of LPCYCLES. The LPSENS command may be phased out.
Availability	Version 1.9 onwards.
Essential when	see LPCYCLES.

Command syntax **lpsens =** *Interval* [*Interval*]

Parameters

Interval is essentially an arbitrary number, ranging from 1 (as slow as possible) to 255 (as fast as possible).

If the machine contains a second loop detector card, a second, different, *Interval* can be set if required to optimise site setup. (If not set assumed to be same for both cards).

Valid range 1 to 255

Default lpsens = 208 208 (192 for loop boards manufactured before September 1993 – check using STATUS) [196 for 83 kHz 4 & 8 loop boards introduced for DBFO contracts]

Examples

lpsens = 50

(one loop card or two loop cards set to the same *Interval*)

lpsens = 50 60

(two loop cards set to different *Intervals.* For example if two 8 loop cards were fitted, loops 1-8 on connector 1 would be set to an *Interval* of 50 and loops 9-16 on connector 2 would be set to an *Interval* of 60).

Display syntax	**lpsens**
Example response	`LPSENS = 128 128`
Keypad menu	`Tech Mode / LPCYCLES`
Related commands	LPCYCLES is now the preferred name for LPSENS. All the other 'LP-' commands can also be used for loop sensor calibration and 'tuning' if required.

Your setting **lpsens =** .....

Your notes

LPSEPS

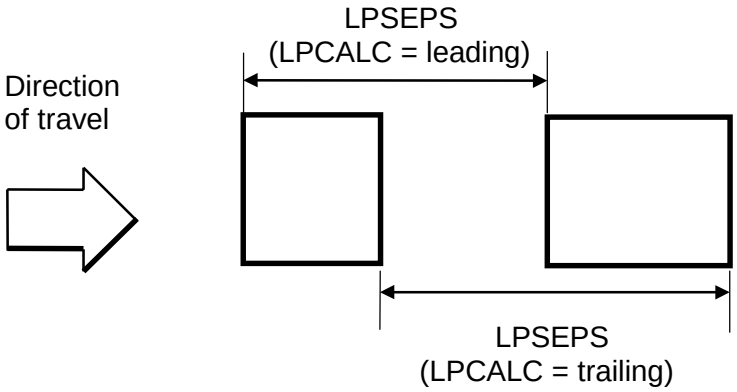
LooP detector SEParationS

Description

This is the separation between the detection loops, measured along the direction of vehicle travel. The loop separation must be set correctly if the calculation accuracy of vehicle speed is important.

The machine measures the time a vehicle takes to travel the distance set by LPSEPS and from this information calculates the speed.

LPSEPS notifies the machine of the separation between the loops in each lane, as measured along the direction of travel. The reference points on both loops depend on whether LPCALC is set to measure from the leading edge of the loops (the default) or from the trailing edge.



Essential when

using loop detectors and INTSPEC involves SPD.

Command syntax **lpseps =** *Separation1* [*Separation2...*]

Parameters

Separation1 is the separation between the loops in lane 1 in the current UNITS of centimetres or inches, measured along the direction of travel as shown above.

If no further values are given, *Separation1* is applied to all lanes.

Separation2 is the loop separation in lane 2, and so on in sequence for all lanes.

Valid range	10 to 999 centimetres, or 4 to 393 inches
Default	lpseps = 400 400 400 400 400 400 400 400 centimetres
	lpseps = 157 157 157 157 157 157 157 157 inches
Example	**lpseps = 402 398 408 400 401 401**
Display syntax	**lpseps**
Example response	`LPSEPS = 480 490 476 400 400 400 400 400`
Keypad menu	`Start Survey / Loop Seps`
Related commands	LPCALC determines where the measurements are made from, as shown above. All the other 'LP-' commands can also be used for loop sensor calibration and 'tuning' if required.

Your setting **lpseps =** .....

Your notes

LPTHRES

LooP detector THREShold

Description LPTHRES sets or displays the threshold sensitivity of the loop detector.

This setting is a compromise. LPTHRES can be decreased if vehicles are not being reliably detected, but this may result in false detection due to noise interference. Conversely, setting LPTHRES too high to reduce false counting may result in vehicles being missed, or in tractor-trailers with high chassis being counted as two separate short vehicles.

To some extent, either of these conditions may be compensated by also 'tuning' LPCYCLES (and as a last resort, LPOVERS and LPUNDERS). When increasing the sensitivity by lowering LPTHRES, you should generally also decrease LPCYCLES to reduce the chance of false triggering.

Essential when using loop detectors and problems are being experienced with noise interference or inadequate sensitivity.

Command syntax **lpthres =** *Threshold* [*Threshold*]

Parameters *Threshold* is the percentage change of inductance required to indicate a vehicle presence *(see Notes).* A higher number implies lower sensitivity. All detectors are set to the same sensitivity.

If the machine contains a second loop detector card, a second, different, *Threshold* can be set if required to optimise site setup. (If not set assumed to be same for both cards).

Valid range 0 to 255

See *Notes* overleaf.

Default lpthres = 50 50 (0.50%) (8 for loop boards manufactured before September 1993 – check using STATUS) [50 for 83 kHz 4 & 8 loop boards introduced for DBFO contracts]

Notes

For Firmware versions 1.90 or earlier, the new default value of 50 still appears as ‘8’ after a restart has been performed, but any change subsequently made using LPTHRES will be correctly displayed.

➲ **CAUTION**

With Marksman 660 *Firmware versions 1.90 or earlier, even if the present value already looks correct, you must still set it manually to the value required.*

For loop boards dated before September 1993 (check using STATUS) the valid range of LPTHRES represents an arbitrary ‘count’ value.

For loop boards manufactured since September 1993, the valid range of LPTHRES represents a change in signal level, from zero to 2.55%.

Examples

lpthres = 30

(one loop card or two loop cards set to the same *Threshold*)

lpthres = 40 50

(two loop cards set to different *Threshold.* For example if two 8 loop cards were fitted, loops 1-8 on connector 1 would be set to a *Threshold* of 40 and loops 9-16 on connector 2 would be set to a *Threshold* of 50).

Display syntax **lpthres**

Example response `LPTHRES = 45 45`

Keypad menu `Tech Mode / LPTHRES`

Related commands All the other ‘LP-’ commands can also be used for loop sensor calibration and ‘tuning’ if required.

LPOVERS and LPUNDERS can be used in conjunction with LPTHRES.

LPUNDERS

Description

Loop detection is subject to electrical noise which may produce false ‘detect’ signals of short duration. Each loop is scanned repeatedly and a vehicle is only considered to have gone past after a number of successive scans have produced no response.

LPUNDERS sets or displays the number of successive negative responses required to register ‘un-detection’.

For loop boards manufactured since September 1993, LPUNDERS is automatically set to be the same as LPOVERS. It is therefore a redundant command for boards manufactured since this date and should not be used.

Essential when

using loop detectors in electrically noisy environments.

Command syntax

lpunders = *Number* [*Number*]

Parameters

Number is the number of successive scans that must find a ‘no response’ condition in order for ‘un-detection’ to be recorded.

If the machine contains a second loop detector card, a second, different, *Number* can be set if required to optimise site setup. (If not set assumed to be same for both cards).

Valid range

1 to 15

Note

Do not set to less than 2

Default

lpunders = 3 3

Examples

lpunders = 8

(one loop card or two loop cards set to the same *Number*)

lpunders = 9 12

(two loop cards set to different *Numbers.* For example if two 8 loop cards were fitted, loops 1-8 on connector 1 would be set to a *Number* of 9 and loops 9-16 on connector 2 would be set to a *Number* of 12).

Display syntax **lpunders**

Example response `LPUNDERS = 9 12`

Keypad menu `Tech Mode / LPUNDERS`

Related commands LPOVERS should be set to the same value(s) as LPUNDERS.

For loop boards manufactured since September 1993, LPUNDERS is automatically set to be the same as LPOVERS and LPUNDERS should not be used.

Your setting **lpunders =** .....

Your notes

MEMFULL

Action when MEMory FULL

Description

There are two limits to the storage capacity of a *Marksman* machine: memory capacity and number of files. The memory capacity depends on the memory card fitted (check using STATUS). The maximum number of files is 20 in a standard machine or 40 with expanded memory.

When either of these two limits is about to be exceeded, you have two choices about what the machine does next. Either it can keep the information it has and stop recording, or it can progressively overwrite the oldest data. MEMFULL changes or displays the current choice.

Essential when

a large amount of data is to be collected, memory capacity is limited, and/or there will be a long interval before the next download of data.

Command syntax **memfull = stop / overwrite**

Parameters

stop will not stop immediately the memory is full, but will attempt to free more space by deleting files marked as Retrieved, starting with the oldest. When this is no longer successful, the machine will stop and all detector cards will be switched off to preserve battery life.

overwrite will delete files and individual data recordings to make space available for new data, in the following order:

1. Retrieved files, oldest file first (as in *Stop*)
2. Closed but un-retrieved files, oldest first
3. When all closed files have been deleted, recordings in the current interval or VBV file will be deleted, oldest recordings first, as the new interval or VBV records are added.

Default

memfull = stop

Example 1

memfull = overwrite

Display syntax	**memfull**
Example response	`MEMFULL = Stop`
Keypad menu	`General Setup / Mem Full Action`

Your setting **memfull =** .....

Your notes

MONITOR

Description MONITOR gives real-time indications of traffic sensor activation, or of carbon monoxide and temperature levels.

Essential when checking sensor performance against vehicle observations, or against independent CO/temperature measurements.

Command syntax **monitor** *LaneNumber* / **gas** / **tcw**

Parameters *LaneNumber* selects the traffic sensor to be monitored.

gas selects CO and temperature monitoring.

tcw selects Tyre Contact Width sensor monitoring (*Marksman 410* only).

Examples of output are shown below. To end the monitoring display, press **Esc** or **Enter**.

Example 1 **monitor 3**

Response `Sensor Check L03 : a`

Explanation Lane 3 is being monitored. If the SENSORS setting involves axle sensors (tube or piezo) the display will show the letter a as above. Each time the sensor is activated, the a briefly becomes a capital `A`.

Other possible codes are (`l` / `L`) for loops and (`w` / `W`) for weigh-in-motion sensors.

Layouts involving multiple sensors will give corresponding displays, e.g. a tube-loop-tube (TLT) layout will show:

`Sensor Check L03 : a  l  a`

Example 2	**monitor gas**
Response	`Temp. xx°C CO yy (Ave.=zz)`
Explanation	The display shows the current temperature reading (*xx*), the current CO reading (*yy*) and the CO reading averaged over the recording interval (*zz*). At the end of the interval, *yy* and *zz* will be the values stored. If no temperature and gas sensor is fitted, *xx*, *yy* and *zz* are replaced by `N/A`. Temperature is measured inside the *Marksman 660*, primarily for sensor compensation. Recorded temperatures will generally track the external ambient temperature, but with a time-lag.
Example 3	**monitor tcw** (*Marksman 410* only)
Response	`TCW = TCW1:Noise1:Thres1 TCW2:Noise2:Thres2`
Explanation	The display shows the last tyre contact width for both sensors (*TCW1*, *TCW2*), the current TCWNOISE value for both sensors (*Noise1*, *Noise2*), and the current TCWTHRES value for both sensors (*Thres1*, *Thres2)*. See TCWCALIB, TCWNOISE and TCWTHRES for further details.
Keypad menu	See Figures 2 and 3, page 20 onwards.

Your notes

OSP, OSP+

On-Site Printout

Description

OSP produces an On-Site Printout of data, configured by the OSPFILTER setting.

OSP+ will produce a similar printout which also shows the time-tagged event (TTE) sensor readings as they are processed.

On-Site Printout continues until the time limit set by OSPTIME, the vehicle count limit set by OSPVEH, or you press **Esc** or **Ctrl-C**.

Essential when

the vehicle-by-vehicle data or TTE sensor readings need to be checked against the actual events at the site, or 'real-time' output is required via a remote communications link.

Command syntax

osp [*Format*]

osp+ [*Format*]

Parameters

Format is one of the standard format numbers from the lists overleaf.

The corresponding OSP+ formats include an additional header line, and include time-tagged events between vehicle printouts.

See Appendix 2 for examples of OSP and OSP+ output formats.

Formats for Printing	*1*	One-line Format: Vehicle Number, Date, Time and full available vehicle details. 160 columns maximum.
	2	Comma and Quote Delimited version of format 1
	3	Three-line Format: Vehicle Number, Date, Time, and all available vehicle measurements, 80 columns maximum. Normally two lines, but three if Weigh-in-Motion data are included; only one line if there are no axle sensors.
	4	Enforcement Graphic Format: Time, Lane, Gap, Speed, Axles, Class, Diagram of vehicle along its length. Maximum 80 columns.

Default format One-line Format 1

Note When using OSP+ at a busy site, where several events are being processed almost simultaneously, some TTE readings may not be listed in quite the order expected.

Examples **osp 3**
osp+ 3

Explanation Vehicle by Vehicle format 3 will be produced. The OSP+ options will also display time-tagged events.

Keypad menu Not available

Related commands OSPFILTER controls the data specification for vehicles to be listed.

OSPTIME sets a time limit for On-Site Printout.

OSPVEH sets the number of vehicles after which On-Site Printout will end.

PRUNITS controls the units for printout (independently from UNITs for recording).

OSPFILTER

On-Site Printout FILTER

Description GRPS allows one logical criterion to be applied when selecting vehicle data to be displayed in On-Site Printout using the OSP or OSP+ commands. OSPFILTER specifies this criterion, or displays the current setting.

OSPFILTER does not affect the additional display of time-tagged event (TTE) sensor readings when using OSP+.

Essential when selective output is required in On-Site Printout.

Syntax **ospfilter = all** / *Condition*

Parameters

all disables On-Site Printout data filtering and displays all vehicles.

Condition is a single logical condition that a vehicle must meet in order to be displayed. This format of this condition is:

keyword comparator value

The valid keywords are:

ACT	Axle count
ASP	Axle separation (largest value for the vehicle)
AWT	Axle weight (largest value for the vehicle)
CLS	Classification
DRN	Direction of travel (1 = normal direction, 2 = reverse direction)
GWT	Gross weight
LAN	Lane number
LEN	Overall length
SPD	Speed
WBT	Overall wheelbase

The valid comparators for use with keywords are:

>	is greater than (but is not equal to)
>=	is greater than, or is equal to
=	is equal to (use only with ACT, CLS, DRN or LAN)
<=	is less than, or is equal to
<	is less than (but is not equal to)
<>	is not equal to (use only with ACT, CLS, DRN or LAN)

See the examples below for typical applications.

Default ospfilter = all

Example 1 **ospfilter = spd > 120**
Explanation Display only vehicles with speed greater than 120, in the current UNITS of speed

Example 2 **ospfilter = cls = 2**
Explanation Display only vehicles in class 2

Display syntax **ospfilter**

Example response `OSPFILTER = ACT < 3`
Explanation Display only vehicles with 2 axles

Keypad menu Not available

Related commands OSP initiates On-site Printout and specifies the output format. OSP+ provides additional time-tagged event listing.

OSPTIME sets a time limit, and OSPVEH the number of vehicles, after which On-Site Printout will end.

INTFILTER, TTEFILTER and VBVFILTER all follow the same logic as OSPFILTER.

Your setting **ospfilter =** .....

OSPTIME

On-Site Printout TIME limit

Description	Once initiated by the OSP or OSP+ commands, On-Site Printout of vehicle data or time-tagged event data continues until the time limit set by OSPTIME (or OSPVEH; or until the operator presses **Esc** or **Ctrl-C**).
Availability	Version 1.9 onwards.
Essential when	On-Site Printout is taking place and the machine is unattended, or when unlimited continuous output might cause remote communications links to 'hang'.

Command syntax	**osptime =** *TimeLimit*
Parameters	*TimeLimit* is the time, in minutes since the issue of the OSP or OSP+ command, after which On-Site Printout will end automatically. To disable OSP timeout, set *TimeLimit* to zero.
Valid range	0 to 999 minutes
Default	osptime = 999
Notes	On-Site Printout can be terminated at any time by pressing **Esc** or **Ctrl-C**.
Example	**osptime = 10**
Display syntax	**osptime**
Example response	`OSPTIME = 30`
Keypad menu	Not available

Related commands

OSP initiates On-site Printout and specifies the output format. OSP+ provides additional time-tagged event listing.

OSPFILTER controls the data specification (as distinct from the format of the printout).

OSPVEH sets the number of vehicles after which On-Site Printout will end.

Your setting **osptime =** .....

Your notes

OSPVEH

On-Site Printout VEHicles limit

Description	Once initiated by the OSP or OSP+ commands, On-Site Printout of vehicle data or time-tagged event data continues until the time limit set by OSPVEH (or OSPTIME; or until the operator presses **Esc** or **Ctrl-C**).
Availability	Version 1.9 onwards.
Essential when	On-Site Printout is taking place and the machine is unattended, or when unlimited continuous output might cause remote communications links to 'hang'.

Command syntax	**ospveh =** *VehicleLimit*
Parameters	*VehicleLimit* is the number of vehicles counted since the issue of the OSP or OSP+ command, after which On-Site Printout will end automatically. To disable this function, set *VehicleLimit* to zero.
Valid range	0 to 9999 vehicles
Default	ospveh = 9999
Notes	On-Site Printout can be terminated at any time by pressing **Esc** or **Ctrl-C**.
Example	**ospveh = 1000**
Display syntax	**ospveh**
Example response	`OSPVEH = 5000`
Keypad menu	Not available

Related commands

OSP initiates On-site Printout and specifies the output format. OSP+ provides additional time-tagged event listing.

OSPFILTER controls the data specification (as distinct from the format of the printout).

OSPTIME sets a time limit for On-Site Printout.

Your setting **ospveh =** .....

Your notes

PAGELEN

PAGE LENgth

Description The ALL or HELP command produces a very long listing, which will scroll off the screen of a typical PC terminal. PAGELEN controls the number of lines of text after which the output pauses with a prompt of:

```
------ More ------
```

Availability Version 1.9 onwards.

Essential when using ALL or HELP with PC terminal software that does not have its own pause or scroll-back feature.

Command syntax **pagelen =** *Lines*

Parameters *Lines* is the number of lines output after which the `More` prompt appears. Press **Enter** to continue to the next `More` prompt, and so on to the end of the listing.

A setting of 0 produces continuous output.

Valid range 0 to 50 lines

Default pagelen = 0

Example **pagelen = 18**

Display syntax **pagelen**

Example response `PAGELEN = 23`

Keypad menu Not available

Related commands PAGELEN only affects output from the ALL or HELP command.

Your setting **pagelen =** .....

Your notes

PASSWORD

Description

Using the INTERFACE command, the remote serial interface of the *Marksman* can be set to one of three modes of security protection:

Read-write Full access to all commands and functions. When files are retrieved via the modem link in this mode, their status is changed from 'Un-retrieved' to 'Retrieved'.

Read-only The user can retrieve files and display the machine status but cannot alter any settings. Any files retrieved in this mode will remain marked as 'Un-retrieved', so that further users can use commands such as **retrieve u** to retrieve the same files.

Locked The serial port gives access to no functions except INTERFACE to change the protection mode.

The INTERFACE command requires a password to authorise the change to **Read-write** or **Read-only** modes. Each of these two modes has its own password, set by the PASSWORD command. (No password is needed when changing to **Locked** mode.)

The PASSWORD command itself can only be used while the machine is in **Read-write** mode.

Availability Version 1.9 onwards.

Essential when setting-up security protection for the remote serial port. Although the password can be omitted, the system then offers no effective protection.

Command syntax **password =** *RWpassword RWpassword*
[ROpassword ROpassword]

Parameters *RWpassword* is the password that must be used with the **interface = rw** command to change to **Read-write** mode from either **Read-write** or **Locked** mode. The *RWpassword* must be repeated for confirmation.

	ROpassword is the optional password for use with **interface = ro**. If omitted, the machine can be changed into **Read-only** mode from either **Read-write** or **Locked** mode without a password. The *ROpassword* must be repeated for confirmation. **password =** will remove all password protection.
Valid range	Up to 8 characters (case-sensitive with no included spaces)
Note	Passwords are case-sensitive, so 'PASSWORD', 'password', 'Password' and 'PassWord' are all different.
Default	password =
Example 1	**password = daylight daylight chestnut chestnut**
Example 2 *Explanation*	**password = daylight daylight** Only Read-write mode will be password-protected.
Display syntax	**password**
Example response	`PASSWORD = LKJ4H LKJ4H F47VV F47VV`
Keypad menu	Not available.
Related commands	INTERFACE sets the remote serial interface to **Read-write**, **Read-only** or **Locked** mode. TIMEOUT sets the interval after which the remote serial interface will revert to **Locked** mode if not used.

Your setting	*You are recommended **not** to write passwords in this manual, but in a safer and less public place.*

PEAKINT

PEAK-period INTerval

Description

It is often useful to collect more detailed data during periods of peak traffic. PEAKINT controls the recording interval during the peak periods designated using PEAKTIME.

See INTERVAL for a more detailed description of recording intervals. Typical PEAKINT values are 5, 10 or 15 minutes. PEAKINT must always be an exact sub-multiple of the INTERVAL setting, which in turn must be an exact sub-multiple of the interval between file breaks (if BREAK is being used). Therefore the three settings should be made in reverse order, starting with BREAK.

Command syntax **peakint =** *IntervalTime*

Parameters *IntervalTime* is the length of the peak-period recording interval in minutes.

Valid range Any exact sub-multiple of the INTERVAL setting, between 1 and 60 minutes (see INTERVAL and BREAK for further rules that apply to interval settings).

Default peakint = 5

Notes PEAKINT is allowed to be equal to INTERVAL, but in that case there is effectively no peak-time recording.

Example **peakint = 10**
Explanation Compatible with INTERVAL periods of 10, 20, 30... minutes, and any allowable BREAK settings.

Display syntax peak**int**

Example response `PEAKINT = 10`

Keypad menu `Start Survey / Peak Interval`

Related commands BREAK sets the time between automatic file breaks.

INTERVAL sets the normal recording interval outside of peak periods.

PEAKTIME controls when peak-period recording begins and ends.

Your setting peak**int =** .....

Your notes

PEAKTIME

PEAK-period recording TIMEs

Description	PEAKTIME controls the times of day when peak-period recordings will start and finish. During peak-period times, the recording interval is controlled by PEAKINT rather than INTERVAL.
Essential when	using peak-interval recording.

Command syntax	**peaktime = off** / [*StartTime1 EndTime1 ...*]
Parameters	*StartTime1* is the first time of day at which peak-period recording is to start, in the 24-hour format 'hh:mm'. *EndTime1* is the time at which the first peak period is to end, in the 24-hour format 'hh:mm'. Each time must be separated from the next by one or more spaces, and times are read in pairs. Further pairs may be added to a maximum of 3. **off** will cancel any previous settings.
Valid range	00:00 to 24:00, each entry (an input of 24:00 will be interpreted and stored as 23:59:59). *StartTime* must be before *EndTime*. All times must be synchronous with the settings established by BREAK, INTERVAL and PEAKINT.
Default	peaktime = off
Example	**peaktime =** 08:00 09:00 11:30 14:00 17:00 19:00
Display syntax	**peaktime**
Example response	`PEAKTIME = 07:30 09:30 17:00 19:00 00:00 00:00`
Keypad menu	`Start Survey / Peak Periods`
Related commands	BREAK sets the time between automatic file breaks.

INTERVAL sets the normal recording interval outside of peak periods.

PEAKINT controls the interval during peak-period recording.

Your setting **peakint =** .....

Your notes

PRINT

Description

Lists a data file from the machine's memory, in ASCII text format. The file is automatically closed and a continuation file opened with an incremented number in the filename extension.

The output need not be literally printed. For example it can be listed to a PC terminal (preferably using software that provides a scroll-back buffer for review), or generated in comma-delimited form for use in a spreadsheet.

A variety of print formats are available (see list below). PRUNITS can specify the units for printout independently of the UNITS setting used for data recording.

Output can be stopped by pressing either **Esc** or **Ctrl-C**.

Command syntax **print =** [*Format*] *Filename.Ext*

Parameters

Format is one of the standard format numbers from the lists below.

Filename.Ext is a valid MS-DOS/GRPS filename (up to 8 characters) followed by a period (.) and an extension of up to three characters. The PRINT command allows only one specific filename at a time.

Formats for Printing Interval Files

1	Date, Time, Channel, and 12 counts. May be selected for single classification counts, up to 13 classes.
2	Date, Time, Channel, and up to three Classifications. Width depends on number of Classifications.
3	Comma and Quote Delimited version of INT-2

Formats for Printing Vehicle by Vehicle Files	*1*	One-line Format: Vehicle Number, Date, Time and full available vehicle details. 160 columns maximum.
	2	Comma and Quote Delimited version of VBV-1
	3	Three-line Format: Vehicle Number, Date, Time, and all available vehicle measurements, 80 columns maximum. Normally two lines, but only one line if there were no axle sensors. A total of three lines if Weigh-in-Motion data are included.
	4	Enforcement Graphic Format: Time, Lane, Gap, Speed, Axles, Class, Diagram of vehicle along its length. Maximum 80 columns.

Default format 2 for Interval files
1 for Vehicle by Vehicle files

Note When the PRINT process is completed the file attribute will be changed to Retrieved (R) – see CHMOD.

Example **print 1 file1234.i00**
Explanation Since the filename extension '.i00' denotes an Interval file, the format selected is 'Interval 1'.

Keypad menu `Output Data / File Print`

Related commands PRUNITS can specify the units for printout alone, separately from the main UNITS command.

EOLCHARS, EOPCHARS and EOFCHARS allow special characters to be output at the end of each line and page, and at the end of the file.

Your setting **print** .....

PROTOCOL

Description Display or change the file transfer protocol.

The protocol may need to be changed to give the best overall data transfer rate for the circuit concerned, and for the communications software at the remote end of the link.

All the available protocols are of the 'Ymodem' type, and the remote communications software must be configured to match.

Essential when configuring the system for file transfers by modem link.

Command syntax **protocol = ymodem / ymodems / ymodemg**

Parameters

ymodem is the industry-standard 'batch Ymodem' implementation using 1024 and 256 byte block sizes. It has built-in error correction.

ymodems is a version of Ymodem that uses 256 byte blocks only. When line conditions are good, YmodemS is slower than standard Ymodem because more blocks have to be sent. On the other hand, YmodemS may be faster on poor lines when the error-correction system is frequently requiring blocks to be repeated.

ymodemg is the industry-standard implementation known by that name. YmodemG has no built-in error correction and is only suitable for links using error-correcting modems – in which case it may prove fastest.

Default protocol = ymodem

Example **protocol = ymodemg**

Display syntax **protocol**

Example response

```
PROTOCOL = YModem
```

Keypad menu Not available

Related commands COM1 and COM2 control the other communications parameters for serial-line communication.

Your setting **protocol =** .....

Your notes

PRUNITS

PRintout UNITS

Description

PRUNITS can specify the units for printout alone, separately from the main UNITS command.

UNITS allows only ‘metric’ and ‘imperial’ whereas PRUNITS gives several options in each.

UNITS and PRUNITS are completely independent. A file can be recorded in one type of units, but output generated with PRUNITS set to a completely different choice.

Command syntax **prunits =** *UnitType*

Parameters *UnitType* can be one of the following:

i1	mph, inches, 10lb
i2	mph, feet, kips (1000lb)
i3	mph, feet, British tons (2240lb)
i4	mph, feet, 10lb
m1	kph, centimetre,10kg
m2	kph, metre, tonne
m3	kph, metre, 10kg
uk1	mph, centimetre, 10kg
uk2	mph, metre, tonne
uk3	mph, metre, 10kg

Default prunits = m1

Example **prunits = uk2**

Display syntax **prunits**
Example response
```
PRUNITS = KPH-MTR-TONNE
```

Keypad menu `General Setup / Print Units`

Related commands OSP, OSP+ and PRINT generate their listings using the units specified by PRUNITS.

Your setting **prunits =** .....

Your notes

PZFILTER

PieZo detector FILTER

Description	Piezo electric axle detectors typically do not produce one pulse when the axle passes over, but several. To 'debounce' this signal, PZFILTER programs the detector card to define a 'dead time' after the first pulse, during which any further pulses generated by the same axle will be ignored. AXFILTER will perform the same function in software, but with piezo detectors it is better to use the hardware command PZFILTER.
Essential when	piezo counting layouts are in use.

Command syntax	**pzfilter =** *Time*
Parameters	*Time* is the dead-time in milliseconds.
Valid range	10 to 30 milliseconds
Default	pzfilter = 20
Example	**pzfilter = 15**
Display syntax	**pzfilter**
Example response	`PZFILTER = 25`
Keypad menu	`Tech Mode / PZFILTER`
Related commands	AXFILTER performs the same debounce function in software, but should only be used with tube detectors. PZTHRES controls the piezo detector threshold level.

Your setting **pzfilter =** .....

Your notes

PZTHRES

PieZo detector THREShold

Description	PZTHRES controls the detection threshold, which is the level of output from the piezo detector card that is considered a valid detection signal.
Essential when	using piezo detectors.

Command syntax	**pzthres =** *DetectLevel*
Parameters	*DetectLevel* is the detection threshold level in millivolts
Valid range	50 to 750 millivolts
Default	pzthres = 750
Notes	The optimum threshold depends on the model of piezo detector card in use, and on the levels of electrical interference at the site.
Example	**pzthres = 500**
Display syntax *Example response*	**pzthres** `PZTHRES = 650`
Keypad menu	`Tech Mode / PZTHRES`
Related commands	PZFILTER sets the piezo detector debounce time in milliseconds.

Your setting **pzthres =** .....

Your notes

R (**REPEAT**)

Description

This command offers a quick way to repeat – or *almost* repeat – the last command you entered. It makes the machine re-type the last command line, but without 'pressing **Enter**' at the end of the line.

You can then edit the line using the ← key to backspace and delete, and then re-type anything you need to change. Then press **Enter**.

It is obviously quicker to enter this command as R (**Enter**) than typing the word REPEAT.

REM

REMark

Description

The REM statement begins a ‘remarks’ or ‘comment’ line which will be ignored by the machine.

As input, REM can be used to add notes to a GRPS programming listing that may be saved as a file and then ‘played’ into the machine at a later time.

When the machine produces output in response to the ALL/HELP command, any GRPS command that has no stored parameters is given a short description beginning with REM:

```
REM DETOFF - Turns detectors off
```

Essential when

adding notes to a GRPS programming listing.

RESTART

Description

RESTART gives access to the Restart Menu, which can be used to delete all data files and reset many of the machine's GRPS parameters to their default values.

When used with a PC Terminal or via the remote serial link and a modem, the RESTART command should display the Restart Menu.

If there is no response to RESTART, a hardware restart is required. Push the special Restart Connector (part number GR010990) onto either the **Com1** or the **Com2** port of the machine and then remove it. The machine will now respond with the Restart Menu.

When the local keypad is used to restart the machine (see below), the LCD display will show the Restart Menu.

Essential when

one or more machines needs to be reset to a known state.

Command syntax

restart [*SerialNo*]

Parameters

If more than one machine is connected to a network but *SerialNo* is included, only that named machine is affected. If *SerialNo* is not included, all connected machines will respond.

Example 1
restart 5348
Explanation
Restarts only machine number 5348.

Example 2
restart
Explanation
Restarts all networked machines.

Restart Menu

The Restart Menu offers four alternative levels of resetting the machine.

1 **Resume** operation with no other changes

2 **Reset Data**: delete all data files

Continued

3 **Reset All**: complete software restart, resetting all parameters (except those listed below) to their default values, and deleting all data files

4 **Restart All**: complete hardware restart, resetting all parameters (except those listed below) to their default values, and deleting all data files.
The level 4 restart is also generated by using the Restart Connector on an interface port, and will generally clear problems that cannot be resolved by restarting at levels 2 or 3.

Data Loss

A level 1 restart (**Resume**) will not close any open files. However, when it is performed from the local keypad, vehicle data may be lost while the Restart Menu is being displayed.

Level 2, 3 or 4 restart will delete all data files stored in the machine's memory. This includes any open files, which will be closed and deleted, and data recording will cease.

Only the following parameters are unaffected by any level of restart: CLASS, CLOCK, COM1, COM2, DATE, DATEFORM, LANGUAGE, TIME and UNITS.

CAUTION

*Following a **Restart 4** with Marksman 660 firmware versions 1.90 or earlier, you* **must** *manually set the value of LPTHRES to the value required for the loop board fitted. If in doubt, set LPTHRES initially to 50.*

Keypad menu

Hold down the **Esc**, **0** and **Edit** keys together for 1 second to see the Restart Menu.

Related commands

SERIAL and STATUS will give the machine's serial number.

RETRIEVE (DOWNLOAD)

RETRIEVE or DOWNLOAD file(s)

Description Retrieve one or more data files from the machine's memory.

The commands RETRIEVE and DOWNLOAD are alternative names with exactly the same effect.

Essential when memory will shortly become full, and before any further data processing can be done.

Command syntax **retrieve** [*Filename.Ext* / **all** / **new** / **u** / **r** / **ur**]

Parameters

Filename.Ext is a valid MS-DOS/GRPS filename (up to 8 characters) followed by a period (.) and an extension of up to three characters. If the period and extension are omitted, all files with that *Filename* will be retrieved.

all will retrieve all files in the machine's memory. Any files that are open will be closed, and a new file started with an incremented number in the filename extension.

new will retrieve all files that have not been retrieved before (or not successfully) and are thus are currently marked Un-retrieved (U).

u will retrieve all Un-retrieved files that are not currently open.

r will retrieve all Retrieved files that are not currently open.

ur will retrieve all Un-retrieved and Retrieved files that are not currently open.

When used with no parameter, **retrieve** is equivalent to **retrieve new**.

Notes

When using **retrieve u** or **retrieve ur**, remember that any files that are currently open will not be retrieved. Take care not to miss any files because of this.

A file retrieval can be started at any time, but may be slower while the machine is actively recording vehicle data. If the retrieval involves closing any currently open file(s), there will be a brief pause while corresponding new file(s) are opened with an incremented number in the filename extension. After this, further recordings can be made in the newly opened files while file transfer is taking place. However, no further keyboard/keypad commands can be entered until the retrieval process is complete.

The file transfer protocol is set with the PROTOCOL command (except that plain ASCII transfers are done with the PRINT command). After entering the RETRIEVE or DOWNLOAD command you must also to start the same file transfer protocol at the receiving device.

Files are transferred one at a time, un-retrieved files first, followed by previously retrieved files, in both cases in order of date opened.

When each file transfer finishes successfully, the attribute of the retrieved file in the machine will be changed to ‘R’ (unless retrieved via the remote serial interface in **Read-only** mode – see INTERFACE).

Example 1
retrieve a34.v01
Explanation
Retrieves that one file

Example 2
retrieve a34-ps
Explanation
Retrieves all files which match the main filename, e.g. A34-PS.V01 and A34-PS.I02

Keypad menu

Not available

Related commands

DOWNLOAD is an alternative name for this command, with exactly the same effect.

BREAK allows the same files to be easily recovered more than once, without interference between users.

CHMOD changes file attributes to ‘R’ or ‘U’.

DELETE is used to delete files after they have been retrieved from the machine’s memory, and for safety will only operate on files marked as Retrieved (R).

DIR lists all files and their attributes.

INTERFACE sets the security protection mode for the remote serial interface, and affects whether files are marked Retrieved (R) after a successful download.

PRINT is used to display a file in the machine’s memory, and when completed will mark each file as Retrieved (R).

Your notes

SENSORS

Description In order to function, the detection and recording software must be configured according to the sensors attached to the machine. The SENSORS command sets or displays the sensor configuration for each lane.

Essential when programming the machine to a new sensor configuration. You will also need to check AXFACTOR, AXSEPS, LPLENS and LPSEPS.

Command syntax **sensors =** [*Config1*] [*Config2...*]

/ **NONE**

Parameters *Config1* is a sensor code from the following list, applied to lane 1, the nearest to the machine.

Config2 etc. are corresponding sensor codes for lane 2, lane 3, and so on in order. If only one code is given, it applies to all lanes and channels.

sensors = NONE disables all sensors and deletes the stored configuration. **NONE** must only be entered in CAPITALS.

➲ CAUTION

SENSORS is a case-sensitive command. *CAPITAL letters specify a full-width sensor; lower-case letters specify a half-width sensor.*

Notes Gas sensors are not affected by the SENSORS command. They are always ‘on’ unless deliberately disabled by **detoff**.

Valid range Up to eight codes from the lists beginning on page 192. Selections that are not possible with the sensor cards fitted to the machine will produce an error message.

Sensor configurations

Sensors must be defined in strict order as defined below, and **must** be connected to the detector inputs on the

machine in the same order. Vehicle detection and classification will not function correctly otherwise.

The sequence is:

1. Near-side to off-side across the lane; then
2. Along the lane in the normal direction of travel; then
3. Following rules 1 and 2 in each successive lane.

Applying these rules strictly and logically will resolve even the most complicated sensor layout.

Example 1 **sensors = LL**
Response `Please wait....`
Explanation Sets all lanes to Loop-Loop.

Example 2 **sensors = LpL LpL L LpL LpL L**
Response `Please wait....`
Explanation There are six lanes, with a repeated arrangement of loop and half-width piezo sensors.

Display syntax **sensors**

Example response
```
SENSORS = LL LL LL LL LL LL LL LL
Please wait....
```

Keypad menu `Start Survey / Sensors`

Related commands Check the following parameters, any time you alter SENSORS settings:

Counting AXFACTOR

Classification LPSEPS, LPLENS, AXSEPS

DETOFF inhibits all sensors but retains the stored configuration (unlike **sensors = NONE** which deletes the configuration).

Nomenclature for SENSORS command

lower case letters	Half-width sensor
UPPER CASE LETTERS	Full-width sensor
Number	Number of sensors at this position
★*Number*	Width of array in lanes
Sensor types	
T	Tube detector (full width only)
P p	Piezo detector (full or half-width)
L	Loop detector
w	Weigh-in-Motion sensor

Sensor Configurations

Axle Counting Only

T	Tube, full lane width
T★2	Tube, two lanes wide
TTN	Two closely-spaced parallel tubes (10cm apart or less), single lane width
TTN★2	Two closely-spaced parallel tubes (10cm apart or less), two lanes wide
p	Piezo, half lane width
P	Piezo, full lane width
P★2	Piezo, two lanes wide

Vehicle Counting Only	
L	Loop, full lane width
L★2	Loop, two lanes wide
N+1★2	2-lane ‘N+1’ configuration (3 loops)
N+1★3	3-lane ‘N+1’ configuration (4 loops)
N+1★4	4-lane ‘N+1’ configuration (5 loops)
N+2★4	4-lane ‘N+2’ configuration (6 loops)

See the Golden River ***Count and Classification Highway Manual*** for a description of ‘N+’ configurations.

Classification using Tubes and Loops	
TT	Two parallel tubes, single lane width
TT★2	Two parallel tubes, two lanes wide
TLT	Tube, Loop, Tube
LTL	Loop, Tube, Loop
LL	Loop, Loop
2(N+1)★2	Two lane ‘2(N+1)’ configuration (6 loops)
2(N+1)★3	Three lane ‘2(N+1)’ configuration (8 loops)
2(N+1)★4	Four lane ‘2(N+1)’ configuration (10 loops)
2(N+2)★4	Four lane ‘2(N+2)’ configuration (12 loops)

See the Golden River ***Count and Classification Highway Manual*** for a description of ‘N+’ configurations.

Classification using Piezo Detectors	
pp	Two parallel half-width Piezos
PP	Two parallel full-width Piezos
PP★2	Two parallel Piezos, 2 lanes wide
PLP	Piezo, Loop, Piezo (all full width)
pLp	Half-width Piezo, full-width Loop, half-width Piezo
LpL	Full-width Loop, half-width Piezo, full-width Loop

Classification and Weighing using WIM Sensors and WIM Card	
ww	Two parallel half-width WIM sensors
LwL	Loop, half-width WIM sensor, Loop
L2wL	Loop, two half-width WIM sensors, Loop
2wL2w	Two half-width WIM sensors, Loop, two half-width WIM sensors
LwLw	Loop, half-width WIM sensors; repeated in next lane
L2wL2w	Loop, two half-width WIM sensors; repeated in same lane

Your setting **sensors =** .....

Your notes

SERIAL

SERIAL number

Description Displays the serial number of the machine. This is required for commands such as RESTART which can be applied selectively when several machines are connected together in a network.

Availability Version 1.9 onwards.

Display syntax **serial**

Example response `SERIAL = 50144`

Keypad menu Not available

Related commands RESTART can address an individual machine by its serial number.

STATUS also gives the serial number as part of the machine's response.

Your setting **serial =** .....

Your notes

SETUP

Description SETUP has two options: **setup general** or **setup survey**. In each case it starts a dialogue to offer each essential item for review, so that you can change the parameters if necessary.

Essential when changing the machine to a completely new use. SETUP ensures that all necessary items have been checked.

Command syntax **setup general / survey**

Parameters **Setup general** covers the items listed below.

Each item shows its current setting followed by an S> prompt. To change a setting at the input prompt, enter a new command using the appropriate Command syntax given in this manual. To leave a setting unchanged and move on to the next item, simply press **Enter** in reply to the prompt.

The following settings are examples only.

```
LANGUAGE = English
UNITS = METRIC
PRUNITS = KPH-MTR-TONNE
CLASS = EUR13
MEMFULL = Stop
DATEFORM = DD/MM/YY
SUMMER = Off
WINTER = Off
BREAK = Off
EOLCHARS = 13 10
EOPCHARS = 60 12
EOFCHARS = 0
```

The items listed in response to **setup survey** will depend on the current settings of SENSORS, INTSPEC and PEAKTIME. You will only be prompted for the values that are relevant, so the list below is only an example.

Each item shows its current setting followed by an S> prompt. Edit or accept the settings as described opposite for **setup general**.

```
DATE =
TIME =
FILENAME =
SITE =
LOCATION =
GRIDREF =
HEADINGS =
SENSORS =
AXSEPS =
LPSEPS =
LPLENS =
WTCAL =
WTAUTCAL =
CHANNELS =
INTSPEC =
INTFILTER =
VBVFILTER =
INTERVAL =
PEAKTIMES =
PEAKINT =
ACTBINS =
LENBINS =
SPDBINS =
TIMEBINS =
WTBINS =
INTONOFF =
VBVONOFF =
```

At the end of the **setup general** or **setup survey** sequence the input prompt reverts to its original form, typically `Q>` or `D>`.

Keypad menu

`General Setup /` (various options)
`Start Survey /` (various options)

Related commands

All the commands listed in response to **setup**.

SITE

SITE name or number

Description SITE changes or displays the descriptive name or number of the site. You should use a meaningful name or serial number, as this information will be used by analysis software such as *Showman Plus*.

If no FILENAME has been specified, the machine uses the first three characters of the SITE name are to generate automatic filenames. When using this method, it is particularly important that the first three characters of the SITE name should be meaningful.

Essential when programming the machine to begin surveys at a new site.

Command syntax **site =** *Name*

Parameters Name can be up to 19 alphanumeric characters.

Valid range Characters can include 0 to 9, A to Z, a to z or '_' (underline), but all alphabetic text will be converted to CAPITALS.

Example 1 **site = cold_lane**
Explanation Spaces are not allowed.

Display syntax **site**

Example response `SITE = COLD_LANE`
Explanation Text has been converted to CAPITALS

Keypad menu `Start Survey / Site Number`

Related commands If no FILENAME has been specified, the machine uses the first three characters of the SITE name are to generate automatic filenames.

Your setting **site =** .....

Your notes

SKEW

Description For Weigh-in-Motion (WIM) the weights of the kerbside and off-side wheels are measured separately using two half-lane-width sensors. Generally the two sensors will register their events at slightly different times; if they are to be considered as a single axle crossing, these events must occur within a certain time 'window'.

Every event is time-tagged with a resolution of 1/8192 seconds, and SKEW sets the time 'window' within which the pair of events must occur in order to be linked.

Availability Version 1.9 onwards.

Essential when setting up weight sensors.

Command syntax **skew =** *Skew*

Parameters *Skew* is the 'window' period in time-tag units of 1/1892 seconds (approximately 8 TTEs per millisecond)

Valid range 1 to 10000 time-tag units (equivalent to 0.12 milliseconds to 1.22 seconds)

Default skew = 240 (equivalent to 30 milliseconds)

Example 1 **skew = 500**

Display syntax **skew**

Example response `SKEW = 320`

Keypad menu `Tech Mode / SKEW`

Your setting **skew =**

.....

Your notes

SPDBINS

SPeeD BINS

Description Vehicles whose speed falls within a specified range can be grouped together in the same 'bin' for counting purposes. SPDBINS specifies the speed ranges.

After defining the SPDBINS, use the SPD parameter in the INTSPEC command to enable speed classification.

Essential when INTSPEC includes SPD.

Command syntax **spdbins =** *LowestBin* [*BinBoundary*...] *HighestBin*

Parameters *LowestBin* is the lowest speed that will be included in the lowest bin. 0 must be supplied if required.

BinBoundary is the lowest speed that will be included in the bin **to the right**.

HighestBin is slightly higher than the highest speed that will be included in any bin (a vehicle with a measured speed exactly equal to *HighestBin* will not be recorded, but this is statistically unlikely) .

Valid range Up to 15 bin boundaries (defining up to 14 bins)

0 to 999 km/h or 621 mph, according to the current setting of UNITS.

Bin boundaries must be entered in increasing order.

Default spdbins = 0 15 30 40 50 60 70 80 90 100 110 120 999 km/h

spdbins = 0 9 19 25 31 37 43 50 56 62 68 75 621 mph

Example **spdbins = 0 10 20 30 40 50 60 70 80 999**
Explanation UNITS are imperial (mph).

Display syntax spdb**ins**

Example response `SPDBINS = 1 20 40 60 80 120 999`

Keypad menu `Start Survey / Bins, Speed`

The display shows the units and the present number of bins.

Related commands INTSPEC is used to select this type of classification.

UNITS specifies the speed units, km/h or mph.

Your setting **spdbins =** .....

Your notes

STARTREC

START RECording

Description STARTREC makes the machine start recording immediately, unless a delayed start has been specified using INTONOFF or VBVONOFF.

The recording will be of the type specified under INTSPEC and/or VBVFILTER. STARTREC always opens a recording file immediately, but if a delayed start has been specified no data will be recorded until the 'start' time is reached.

Essential when a survey has been set up. **No data will be recorded without the STARTREC command.**

Command syntax **startrec** **int** / **vbv** / **both** [*Filename*]

Parameters

int starts an interval recording

vbv starts a vehicle-by-vehicle recording

both starts a combined interval and VBV recording

Filename, if supplied, will replace the previous setting of FILENAME before recordings commence.

Valid range *Filename* must be a valid MS-DOS/GRPS filename (up to 8 characters). Do not include an extension to the filename.

An error message will occur if the specified type of recording is already in progress, or if there are already 100 files in data memory with the current filename prefix.

Default The current FILENAME will be used if none is specified with STARTREC.

Example **startrec int file1234**

Keypad menu	`Start Survey / Start Logging`
Related commands	INTSPEC and VBVFILTER specify the characteristics of the recording to be started. INTONOFF and VBVONOFF can delay the start of data recording. FILENAME specifies the filename that will be used as a default unless overridden in STARTREC. STOPREC stops recordings.

Your notes

STATUS

Description STATUS displays the machine's hardware configuration and carries out internal checks.

Display syntax **status**

Example response (Details vary with machine configuration, date, time etc. The layout may also change in future firmware revisions.)

```
Marksman 660 Version 1.98 (20/02/98)
Hardware issue 3.00
Serial 12345
Clock = 14:51:58  20/02/98
Battery =  6.4 Volts
The following detector cards are installed
Slot  Board                                    Version  Date
----  ---------------------------------------  -------  ----
1     8 Loop                                  1.3  12 Apr 96
3     8 Loop                                  1.3  12 Apr 96
Keypad and LCD fitted : YES
Free Memory Check  : PASSED

Processor Loading Factor : 46
Memory Logging : 190192 Bytes Total  99.71%  189632 Bytes Free
       Program : 11728 Bytes
Heap Check     : PASSED
Languages available : English
```

Keypad menu Press **Esc - Esc - Esc**.

Related commands ALL/HELP gives the software configuration.

SERIAL displays the machine's serial number.

Your notes

STOPREC

STOP RECording

Description

STOPREC makes the machine stop the specified type(s) of recording immediately, and closes all relevant files.

STOPREC overrides any delayed stop that has been specified using INTONOFF or VBVONOFF.

Command syntax **stoprec** [**int** / **vbv** / **both**]

Parameters

int stops an interval recording

vbv stops a vehicle-by-vehicle recording

both stops all recording

stoprec with no parameters is equivalent to **stoprec both**

Example 1 **stoprec int**

Keypad menu `Stop Survey / Stop Logging`

Related commands INTONOFF and VBVONOFF can delay the start of data recording, but STOPREC (with the appropriate parameter) will override either or both.

STARTREC starts recordings.

Your notes

SUMMER

Begin SUMMER time

Description SUMMER controls the date when a Winter/Summer time break is inserted into a recording. The day is always a Sunday.

At 02:00 Winter Time on the Sunday specified by SUMMER, any current files are closed. The clock is then set forward 1 hour and a new file is opened starting at 03:00 Summer Time.

Golden River data analysis software such as *Showman Plus* will handle the transition automatically.

Essential when the programmed recording interval spans the Sunday when Summer Time begins.

Command syntax **summer = 1st / 2nd / 3rd / 4th / last** *mm*
/ off

Parameters **1st / 2nd / 3rd / 4th / last** designate the Sunday in the month.

mm is the month number (one or two digits).

off inhibits the Summer Time adjustment.

The syntax of SUMMER is not affected by the DATEFORM setting.

Default summer = off

Example **summer = 1st 4**
Explanation Change on the first Sunday in April

Display syntax **summer**

Example response `SUMMER = 3rd 3`

Keypad menu `General Setup / Summer`

Related commands WINTER controls the date of the Summer/ Winter Time transition.

Your setting **summer =** .....

Your notes

SWINPUT

SWitch INPUT

Description Two types of Switch cards are available: the Switch Input/Output card (GR6683) and the Tube/Switch Input cards (GR6601 and GR6632). If the machine is being used with external axle or loop detector(s) connected to the input port(s) of either type of card, SWINPUT determines what kind of event will be recorded by a switch closure.

Essential when the machine is being used with external axle or loop detector(s) connected to the input port(s) of a Switch Input/Output card or a Tube/Switch Input card.

Command syntax **swinput = none / axle / loop**

Parameters **none** ignores all switch inputs.

axle treats all switch closures at the input port(s) as axle detections.

loop treats all switch closures at the input port(s) as loop detections.

Default swinput = none

Example **swinput = loop**

Display syntax **swinput**

Example response `SWINPUT = AXLE`

Keypad menu `Tech Mode / Switch Input`

Related commands SWOUTPUT selects which events will activate a switch output card. If SWOUTPUT is being used, SWINPUT must be set to **none** (the default value).

Your setting **swinput =** .....

Your notes

SWOUTPUT

SWitch OUTPUT

Description

When using the Switch Input/Output card (GR6683), SWOUTPUT applies the same type of logic as the 'xxxFILTER' commands to control which vehicle detections are allowed to activate the switches on each of the output ports of the card.

Each time the stated condition is satisfied, the specified switch contact is closed for between 0.5 and 1.5 seconds.

Essential when

external contacts are to be operated by the machine.

Command syntax

There are two alternative forms of command syntax. You can use either one, but not both at the same time.

swoutput *SwNumber* **= all** / *Condition* / **none**

swoutput = on / off

Parameters

In the first from of command syntax, *SwNumber* is the number of the switch output port that is being controlled.

all enables the switch output port selected by *SwNumber*, for all vehicles.

Condition is a single logical condition that a vehicle must meet in order to be recorded. This format of this condition is:

keyword comparator value

The valid keywords are:

ACT	Axle count
ASP	Axle separation (largest value for the vehicle)
AWT	Axle weight (largest value for the vehicle)
CLS	Classification

Continued

DRN	Direction of travel (1 = normal direction, 2 = reverse direction)
GWT	Gross weight
LAN	Lane number
LEN	Overall length
SPD	Speed
WBT	Overall wheelbase

The valid comparators for use with keywords are:

>	is greater than (but is not equal to)
>=	is greater than, or is equal to
=	is equal to (use only with ACT, CLS, DRN or LAN)
<=	is less than, or is equal to
<	is less than (but is not equal to)
<>	is not equal to (use only with ACT, CLS, DRN or LAN)

See the examples below for typical applications.

none disables operation of the switch output port selected by *SwNumber*

Alternative form

The alternative form of SWOUTPUT command syntax controls all the switch output ports together:

swoutput = on / off

on enables all switch output ports using any pre-existing *Condition* settings.

off inhibits all switch output ports but still stores any existing *Condition* settings.

Valid range

The output port *SwNumber* must correspond to the correct set of terminals on the switch card, and the *Condition* must be compatible with the UNITS setting and the sensors available.

Defaults	swoutput = off
Example	**swoutput 8 = cls = 13**
Explanation	Contact 8 closes when class = 13
Display syntax	**swoutput** *SwNumber*
Parameters	*SwNumber* is the number of the switch output port that is being interrogated.
Example response	`SWOUTPUT 1 = SPD > 55`
Keypad menu	`Tech Mode / Switch Output`
Related commands	SWINPUT determines what kind of event will be recorded by an input to the Switch Input/Output card (GR6683) or a Tube/Switch Input card (GR6601 or GR6632), if the machine is being used with external axle or loop detector(s). SWINPUT must be set to **none** (the default setting) if switch outputs are in use. INTFILTER, OSPFILTER, TTEFILTER and VBVFILTER all use the same type of logic as SWOUTPUT.

Your setting **swoutput =** .....

Your notes

TCWCALIB

Tyre Contact Width CALIBration ***Marksman 410* only**

Description Starts calibration procedure for Tyre Contact Width tube sensors, to discriminate between bicycles and larger vehicles.

Command syntax **TCWCALIB**

Notes The function of this command is to automatically set (calibrate) the noise and threshold parameters by monitoring bicycles passing over the sensors.

When this command is used, the machine goes into calibration mode. To calibrate the sensors using the PC terminal, perform the following steps:

1. Type **tcwcalib** and press the **Enter** key.
2. Press the **+** key just before a bicycle passes over the sensor(s).
3. Press the **–** key just after the bicycle has passed over the sensor(s).
4. Repeat steps 2-3 for at least five bicycles.
5. Press **Enter** to end calibration. This automatically sets the noise and threshold parameters, if five or more bicycles have been logged.

If five or more bicycles have been logged, the new bicycle parameters for each sensor are calculated as follows:

TCWNOISE = The smallest bicycle TCW reading, minus 5.

TCWTHRES = The largest bicycle TCW reading, plus 5.

You can adjust these two commands further if necessary; see the TCWNOISE and TCWTHRES command references for details.

➲ CAUTION

No motor vehicles may cross over the sensors during the calibration process. If a motor vehicle was present on the sensors during calibration, you **must** *abort the procedure by pressing the* ***Esc*** *key. Then repeat the calibration.*

Keypad menu

`Tech mode / TCW Calibrate`

Use the ↑ key in place of **+** in the calibration procedure detailed above; use the ↓ key in place of **–**.

Related commands

TCWNOISE adjusts the sensor noise threshold.

TCWTHRES defines the sensor threshold level for classification as a motor vehicle.

Marksman 410 enhancements to CLASS, MONITOR and OSP+.

Your notes

TCWNOISE

Tyre Contact Width NOISE **_Marksman 410_ only**

Description This parameter sets or displays the noise threshold for each Bicycle Discriminating tube sensor. This value is used to determine whether a signal is a valid tyre crossing or sensor noise.

TCWNOISE is automatically set by the TCWCALIB procedure but may also be adjusted manually.

Command syntax **tcwnoise =** *Noise-threshold1 [Noise-threshold2]*

Parameters *Noise-threshold1* is the noise threshold for the first sensor. *Noise-threshold2* is the noise threshold for the second sensor, if used.

Valid range 0 to 255

Default tcwnoise = 1

Example **tcwnoise = 10 8**

Display syntax **tcwnoise**

Example response `TCWNOISE = 12 15`

Keypad menu `Tech Mode / TCW Noise`

Related commands TCWCALIB starts sensor calibration.

TCWTHRES defines the sensor threshold level for classification as a motor vehicle.

Marksman 410 enhancements to CLASS, MONITOR and OSP+.

Your setting **tcwnoise =** .....

TCWTHRES

Tyre Contact Width THREShold ***Marksman 410* only**

Description This parameter defines the threshold level for each Bicycle Discriminating tube sensor. This value is used to determine whether a tyre crossing is due to a bicycle or a motor vehicle.

TCWTHRES is automatically set by the TCWCALIB procedure but may also be adjusted manually.

Command syntax **TCWTHRES =** *Threshold1 [Threshold2]*

Parameters *Threshold1* is the threshold level for the first sensor.
Threshold2 is the threshold level for the second sensor.

Valid range 1 to 255

Default 30

Example **tcwthres = 25 20**

Display syntax **tcwthres**

Example response `TCWTHRES = 33 40`

Keypad menu `Tech mode / TCW Threshold`

Related commands TCWCALIB starts sensor calibration.

TCWNOISE adjusts the sensor noise threshold.

Marksman 410 enhancements to CLASS, MONITOR and OSP+.

Your setting **tcwthres =** .....

TEMP

TEMPerature (formerly WTtemp)

Description Weight indications from a WIMstrip sensor exhibit a linear temperature effect, so by calibrating its output signals using TEMPCAL a WIMstrip can also be used as a temperature sensor.

TEMP will then display the current road temperature sensed by the WIMstrip.

Display syntax **temp**

Example response TEMP = 18

Explanation The display is in °C, truncated (not rounded) to a whole number.
Unless the temperature calibration has been set using both **tempcal high** and **tempcal low** (see TEMPCAL), the response will be `TEMP = Uncalibrated`.

Keypad menu Not available

Related commands TEMPCAL sets the calibration for WIMstrip temperature sensing.

Your notes

TEMPCAL

TEMPerature CALibration (formerly WTtempcal)

Description Weight indications from a WIMstrip sensor exhibit a linear temperature effect, so by calibrating its output signals using TEMPCAL a WIMstrip can also be used as a temperature sensor.

Essential when using a WIMstrip for road temperature measurement.

Command syntax **tempcal high =** *HighTemp [WIMref]*

tempcal low = *LowTemp [WIMref]*

Parameters Both the **tempcal high** and the **tempcal low** setting are required in order to establish the calibration, but either can be entered at any time.

HighTemp and *LowTemp* are the present road surface temperatures, as measured by other means. For greatest accuracy, *HighTemp* and *LowTemp* should be measured and input at times chosen to give the greatest possible temperature difference.

WIMref is an optional technical parameter which overwrites the machine's internally measured WIMstrip reference value (not recommended for customer use). A separate *WIMref* value is stored for each WIMstrip sensor.

Valid range -50 to +70 °C

Default tempcal high 0 0

tempcal low 0 0

Example **tempcal high = 18**

Explanation Road surface temperature, as measured by other means. Decimal values will be truncated (not rounded) to a whole number.

Display syntax **tempcal**

Example response

```
TEMPCAL HIGH = 20 48997 46697 45023 45224 0 0 0 0 0 0 0 0 0 0 0 0
TEMPCAL LOW = 5 48592 46359 44789 44921 0 0 0 0 0 0 0 0 0 0 0 0
```

Explanation The first number in each response line is the temperature, truncated (not rounded) to a whole number. The remaining numbers are the *WIMref* values, one for each WIMstrip sensor in the array (16 in this example).

Unless **tempcal high** and **tempcal low** have both been set, the machine responds to TEMP and TEMPCAL with `Uncalibrated.`

Keypad menu Not available

Related commands TEMP displays the current road temperature sensed by the WIMstrip sensor.

Your setting **tempcal high =** .....

tempcal low = ...

Your notes

TEST

Description TEST is always followed by a parameter, and then performs the appropriate hardware or software sub-system test.

Essential when The present configuration of the machine is to be verified or a hardware fault is suspected.

Command syntax **test display / file / keyboard / lpstatus / sysaudit / vehcount / volts**

---

test display Tests the machine's inbuilt LCD display system.

This test requires an external PC Terminal. The machine goes through its entire character set on the LCD display.

test file Creates an example test file of each type (interval and vehicle-by-vehicle) in the machine's data memory, using the current configuration.

The main use of these files is to test the machine's download functions and external data analysis software.

test file does not necessarily create identical files on each occasion.

test keyboard Tests the machine's inbuilt keypad. This test requires an external PC Terminal.

The machine will prompt you to press each key in order. As you press each key it will copy your input to the external Terminal.

test lpstatus Displays a status value for each loop. The value will be zero if the loop card has not reported any errors for that loop, or otherwise the last error code reported. All error codes are reset to zero when the loops are turned on using the DETON or SENSORS command.

Availability Version 1.95 onwards.

Example response `LPSTATUS = 0 0 0 33 0 0 0 0 0 0 0 0 0 0 0 0`

Explanation The loop card last showed an error 33 on loop 4.

Keypad menu Status / Loop Status

test sysaudit Displays the following audit information about the system configuration:

1. The number of changes made to the *Marksman 660* configuration since it was powered-up with the factory default configuration. Each time the *Marksman 660* accepts a GRPS command used in its **command syntax**, this counts as one change. The exception is CLOCK, which allows the time/date to be routinely synchronized to a remote base station.

   The recorded total will only be lost if the internal backup battery for the *Marksman 660*'s internal clock/calendar becomes discharged.

2. The date of the last configuration change.

3. The 'checksum' of the configuration after it was last changed. Every time the configuration is changed, it will generate a new and different checksum.

4. The current checksum. If this is the same as the previous one, it confirms the integrity of the *Marksman 660*'s present configuration.

Availability Version 1.95 onwards.

Example response `SYSCHECK = 81 28/10/96 19AB 19AB`

test vehcount Displays a count of the vehicles in each lane. This command is intended for use by automatic polling software.

test vehcount with no parameters displays the total count in each lane since the counts were last reset (see below), or since the last RESTART command.

test vehcount = 0 resets all the vehicle counts to zero.

test vehcount = *NewStart* resets all the vehicle counts to the same value *NewStart*, which can be in the range from 0 to 49999.

Availability Version 1.98 onwards.

Example response `TEST VEHCOUNT = 2524 2487 0 0 0 0 0 0`

Explanation Up to eight lanes can be displayed. Counts for inactive lanes are shown as `0`.

Keypad menu Status / Vehicle Count

test volts The machine displays the current battery voltage (also available in the STATUS output).

Example response `TEST VOLTS = 6.4`

Keypad menu TEST commands are only available ‘test lpstatus’ and ‘test vehcount’.

Your notes

TIME

TIME of day

Description	This command will change or display the time held in the machine's internal clock/calendar; but CLOCK is now the preferred command to do this.
Essential when	the clock/calendar needs to be reset (e.g. because both the main battery and the backup battery have been discharged or disconnected).

Command syntax	**time =** *hh***:***mm [***:***ss]*
Parameters	*hh* are the hours in 24-hour clock format, *mm* are the minutes and *ss* the seconds. The colons are required. Include leading zeros and omit spaces.
Valid range	00:00:00 to 23:59:59
Default	Set to the current time when the unit is first powered-up in the factory. TIME is not affected by any resets (not even RESTART 4); all changes must be entered explicitly.
Example	**time = 09:42**
Display syntax	**time**
Example response	`TIME = 14:40:32`
Keypad menu	`Start Survey / Clock`
Related commands	CLOCK sets both the date and time of day, and should be used in preference to DATE and TIME separately. DATE sets the date.

Your notes

TIMEBINS

TIME BINS

Description In Gap and Headway interval recordings, vehicles whose gap/headway falls within a specified time range can be grouped together in the same 'bin' for counting purposes. TIMEBINS specifies the time ranges.

After defining the TIMEBINS, use the GAP and/or HWY parameter in the INTSPEC command to enable the appropriate classification.

Essential when INTSPEC includes GAP and/or HWY.

Command syntax **timebins =** *LowestBin* [*BinBoundary*...] *HighestBin*

Parameters *LowestBin* is the shortest time that will be included in the lowest bin. 0 must be supplied if required.

BinBoundary is the shortest time that will be included in the bin **to the right**.

HighestBin is slightly higher than the longest time that will be included in any bin (a vehicle with a measured gap or headway exactly equal to *HighestBin* will not be recorded, but this is statistically unlikely) .

Valid range Up to 15 bin boundaries (defining up to 14 bins)

0 to 999.9 seconds

Bin boundaries must be entered in increasing order.

Default timebins = 0 5.0 10.0 20.0 999.9 seconds

Example **timebins = 0 1 2 3 5 10 20 30 999.9**

Display syntax **timebins**

Example response

```
TIMEBINS = 0 1.0 2.0 5.0 10.0 999.9
```

Keypad menu `Start Survey / Bins, Length`

The display shows the units and the present number of bins.

Related commands INTSPEC is used to select this type of classification.

Your setting **timebins =** .....

Your notes

TIMEOUT

Description	If the remote serial interface is in **Read-write** or **Read-only** mode (see INTERFACE), TIMEOUT can set a time delay after which that interface will revert automatically to **Locked** mode if no key has been pressed.
Availability	Version 1.9 onwards.
Essential when	security protection of the remote serial port is required to maintain itself automatically.

Command syntax	**timeout =** [*TimeDelay*]
Parameters	*TimeDelay* is the number of minutes after which the remote serial interface will revert to **Locked** mode if no key has been pressed. **timeout = 0** will disable this function.
Valid range	0 to 60 minutes
Default	timeout = 0 (disabled)
Example	**timeout = 10**
Display syntax	**timeout**
Example response	`TIMEOUT = 5`
Keypad menu	Not available
Related commands	INTERFACE sets the remote serial interface to **Read-write**, **Read-only** or **Locked** mode. PASSWORD sets the passwords required to gain access to **Read-write** and **Read-only** modes.

Your setting **timeout =** .....

Your notes

TTEFILTER

Time-Tagged Event FILTER

Description GRPS allows one logical criterion to be applied when selecting time-tagged events (TTEs) to be recorded along with vehicle data in vehicle-by-vehicle files. TTEFILTER specifies this criterion, or displays the current setting.

The TTEFILTER setting is usually the same as that of VBVFILTER, so that the two sets of data correspond.

Essential when selective TTE recording is required in vehicle-by-vehicle files.

Syntax **ttefilter = none** / *Condition*

Parameters **none** disables TTE recording.

Condition is a single logical condition that a vehicle must meet in order for its TTEs to be recorded. This format of this condition is:

keyword comparator value

The valid keywords are:

ACT	Axle count
ASP	Axle separation (largest value for the vehicle)
AWT	Axle weight (largest value for the vehicle)
CLS	Classification
DRN	Direction of travel (1 = normal direction, 2 = reverse direction)
GWT	Gross weight
LAN	Lane number
LEN	Overall length
SPD	Speed
WBT	Overall wheelbase

The valid comparators for use with keywords are:

>	is greater than (but is not equal to)
>=	is greater than, or is equal to
=	is equal to (use only with ACT, CLS, DRN or LAN)
<=	is less than, or is equal to
<	is less than (but is not equal to)
<>	is not equal to (use only with ACT, CLS, DRN or LAN)

See the examples below for typical applications.

Valid range The *Condition* must be compatible with the UNITS setting and the sensors available.

Default ttefilter = none

Example 1 **ttefilter = spd > 120**
Explanation Record only TTEs associated with vehicles that have speed greater than 120, in the current UNITS of speed

Example 2 **ttefilter = cls = 2**
Explanation Record only TTEs associated with vehicles in class 2

Display syntax **ttefilter**

Example response `TTEFILTER = ACT < 3`
Explanation Record only TTEs associated with 2-axled vehicles

Keypad menu Not available

Related commands INTFILTER, OSPFILTER and VBVFILTER all follow the same logic.

Your setting **ttefilter =** .....

UNITS

Description Display or change the units which will apply to many other command settings. Measurements will be made and the data stored accordingly.

In contrast, PRUNITS only affects the output format, and does not change the stored data. Conversion takes place automatically, so you can record a data file with UNITS set to one choice, but generate output with PRUNITS set to any other.

Essential when setting up the instrument for surveys that do not use the default metric units.

Command syntax **units = metric / imperial**

Parameters **metric** units are kilometres/hour, centimetres, kilograms and degrees Celsius

imperial units are miles/hour, inches, pounds and degrees Celsius

Default units = metric
UNITS is not affected by any resets (not even RESTART 4); all changes must be entered manually.

Example **units = imperial**

Display syntax **units**

Example response `UNITS = metric`

Keypad menu `General Setup / Units`

Related commands The following commands are affected by UNITS:

AXSEPS, EOVDACC, EOVDSEP, EOVDSPEED, INTFILTER, LENBINS, LPLENS, LPSEPS, OSPFILTER, SPDBINS, SWOUTPUT, TTEFILTER, VBVFILTER, WTAUTCAL, WTBINS, WTCALIB, WTCALIB+.

PRUNITS is **not** affected by the UNITS setting.

Your setting **units =** .....

Your notes

UPLOAD

UPLOAD file(s)

Description

UPLOAD transfers a file to the machine from the PC Terminal using the transfer method specified by the PROTOCOL command.

An upload can only be started when a survey is not active.

After entering this command you also have to start the upload procedure in the Terminal's communications software, selecting the same protocol as set in the machine.

No further commands can be entered until the upload has finished, or you press **Ctrl-X** to abandon the process.

Essential when

transferring a configuration file to the machine before using CFGLOAD.

Command syntax **upload**

Example response

```
Ready to receive files using YModem
Start local procedure now (press control-x
to abort) . . .
```

Explanation

The protocol stated above will depend on the current PROTOCOL setting. The machine will then wait for the file transfer to be initiated by the Terminal's communications software, or until you press **Ctrl-X** to abandon the process.

Keypad menu Not available

Related commands

DOWNLOAD/RETRIEVE initiates file transfer from the machine to the PC Terminal.

PROTOCOL sets the type of Ymodem file transfer protocol to be used for both UPLOAD and DOWNLOAD/ RETRIEVE.

Your notes

VBVFILTER

Vehicle-By-Vehicle FILTER

Description GRPS allows one logical criterion to be applied when selecting vehicles to be recorded in vehicle-by-vehicle files. VBVFILTER specifies this criterion, or displays the current setting.

VBVFILTER does not affect vehicle selection for On-Site Printout, which is controlled separately by OSPFILTER.

Essential when selective recording is required in vehicle-by-vehicle files.

Syntax **vbvfilter = all** / *Condition*

Parameters

all disables VBV filtering and records all vehicles

Condition is a single logical condition that a vehicle must meet in order to be recorded. This format of this condition is:

keyword comparator value

The valid keywords are:

ACT	Axle count
ASP	Axle separation (largest value for the vehicle)
AWT	Axle weight (largest value for the vehicle)
CLS	Classification
DRN	Direction of travel (1 = normal direction, 2 = reverse direction)
GWT	Gross weight
LAN	Lane number
LEN	Overall length
SPD	Speed
WBT	Overall wheelbase

The valid comparators for use with keywords are:

>	is greater than (but is not equal to)
>=	is greater than, or is equal to
=	is equal to (use only with ACT, CLS, DRN or LAN)
<=	is less than, or is equal to
<	is less than (but is not equal to)
<>	is not equal to (use only with ACT, CLS, DRN or LAN)

See the examples below for typical applications.

Valid range The *Condition* must be compatible with the UNITS setting and the sensors available.

Default vbvfilter = all

Example 1 **vbvfilter = spd > 120**
Explanation Record only vehicles that have speed greater than 120, in the current UNITS of speed

Example 2 **vbvfilter = cls = 2**
Explanation Record only vehicles in class 2

Display syntax **vbvfilter**

Example response `VBVFILTER = ACT < 3`
Explanation Record only 2-axled vehicles

Keypad menu `Start Survey / VBV Filter`

Related commands INTFILTER, OSPFILTER and TTEFILTER all follow the same logic.

VBVONOFF

Vehicle-By-Vehicle recording ON / OFF

Description	VBVONOFF controls the dates and times when vehicle-by-vehicle (VBV) recordings will start and finish.
Essential when	programming an instrument to be left to record data at some later time.

Command syntax **vbvonoff =** *StartTime StartDate EndTime EndDate* / **off**

Parameters

StartTime is the time at which recording is to start, in the 24-hour format ‘hh:mm’.

StartDate is the date on which recording is to start, in the format specified by DATEFORM.

EndTime is the time at which recording is to end, in the 24-hour format ‘hh:mm’.

EndDate is the date on which recording is to end, in the format specified by DATEFORM.

off cancels any previous settings.

vbvonoff is automatically set to **off** when the programmed recording is complete (i.e. *Endtime/EndDate* has passed).

Valid range

00:00 01/01/96 (1996) to 23:59 31/12/95 (2095), or equivalent according to the DATEFORM format.

StartTime/StartDate must be before *Endtime/EndDate*.

An input of 24:00 will be interpreted and stored as 00:00 of the next day. Starting with *Marksman* firmware version 1.94, two-digit year numbers must be 96 or later, in order to be interpreted correctly (as 1996 to 2095). The year can also be entered in four-digit format.

Default vbvonoff = off

➲ CAUTION

After setting VBVONOFF, you must then enter STARTREC VBV to enable vehicle-by-vehicle recording. This will open a file but no data will be entered until StartTime/StartDate *is reached.*

Example **vbvonoff =** 00:00 01/04/98 24:00 30/04/98
Explanation Recording will stop at 00:00 on 01/05/98

Display syntax **vbvonoff**

Example response `VBVONOFF = 00:00 01/04/98 00:00 01/05/98`

Keypad menu `Start Survey / VBV On/Off`

Related commands DATEFORM sets the format for all dates used in GRPS.

STARTREC enables recording from the time set by VBVONOFF.

STOPREC stops or prevents all recording, regardless of the VBVONOFF setting. STOPREC VBV is specific to VBV recording.

Your setting **vbvonoff =** .....

Your notes

WINTER

Begin WINTER time

Description WINTER controls the date when a Summer/Winter time break is inserted into a recording. The day is always a Sunday.

At 02:00 Summer Time on the WINTER Sunday, any current files are closed. The clock is then set back 1 hour, and a new file opened starting at 01:00.

Golden River data analysis software such as *Showman Plus* will handle the transition automatically.

Essential when the programmed recording interval spans the Sunday when Winter Time begins.

Command syntax **winter = 1st / 2nd / 3rd / 4th / last** *mm*
/ **off**

Parameters **1st / 2nd / 3rd / 4th / last** designate the Sunday in the month.

mm is the month number (one or two digits)

off inhibits the Winter Time adjustment

The syntax of WINTER is not affected by the DATEFORM setting.

Default winter = off

Example **winter = 2nd 10**
Explanation Change on second Sunday in October

Display syntax **winter**

Example response `WINTER = LAST 9`

Keypad menu `General Setup / Winter`

Related commands SUMMER controls the date of the Winter/Summer Time transition.

Your setting **winter =** .....

Your notes

WTAUTCAL

WeighT AUTo-CALibration

Description

The machine calculates a rolling average of the weight of the front axle of the vehicles in the specified class, averaged over all lanes. The front axle weight is relatively independent of loading, so this average weight can remain relatively constant over time.

Weight auto-calibration offers the option to use this information to correct for any short-term fluctuation in the calibration of the weight sensors.

If weight auto-calibration is enabled, the sensor calibration will be updated whenever the newly computed average differs from its previous value by more than 1%.

Essential when

environmental conditions make the weight sensor calibration subject to fluctuations. However, there must be enough vehicles of the selected class to make the average front-axle weight less variable than the sensor calibration itself.

Command syntax

wtautcal = on [*Class* *StartWeight*]

/ **off**

Parameters

Class is any valid number within the classification scheme being used.

StartWeight is the long-term average front-axle weight (in the current weight UNITS), determined from previous surveys or by some other means. Sensor calibration will be altered to keep the ongoing average at this value (±1%).

wtautcal off inhibits auto-calibration but continues to store the previous setting (or the default setting).

wtautcal on will then continue from its previous settings.

Valid range

Any values

Default

off 9 4082 (kilograms)

off 9 9000 (pounds)

Notes Whenever an automatic re-calibration takes place, a `WTAUTCAL` line is written in the active VBV file using the same format as if the command has been entered manually. Weight calibration factors are also revised and a `WTCAL` line inserted.

Example **wtautcal = on 9 5920**
Explanation If units are kilograms, sensor calibration will be altered when necessary to keep the ongoing average weight for the front axles of vehicle class 9 at 5920kg ±1%.

Display syntax **wtautcal**

Example response `WTAUTCAL = OFF 8 4370`
Explanation The previous setting is stored, but not currently in use.

Keypad menu Not available

Related commands WTAUTCAL updates the calibration factors by using the syntax of WTCAL.

Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtautcal =** .....

Your notes

WTBINS

WeighT BINS

Description

When using Weigh-in-Motion, vehicles having different axle or gross weights can be grouped together in the same 'bin' for counting purposes. WTBINS controls this grouping.

This method of classification should be used in a region where there is no standard for vehicle classification. After defining the WTBINS, use the AWT and/or GWT parameter in the INTSPEC command to enable Weigh-in-Motion.

Essential when

INTSPEC includes AWT and/or GWT.

Command syntax

wtbins = *LowestBin* [*BinBoundary*...] *HighestBin*

Parameters

LowestBin is the lowest weight that will be included in the lowest bin.

BinBoundary is the lowest weight that will be included in the bin **to the right**.

HighestBin is slightly more than the highest weight that will be included in any bin.

Valid range

Up to 15 bin boundaries (defining up to 14 bins)

Weights from 0 to 294833 kg, 649990 pounds

Weights must be entered in increasing order.

Default

wtbins = 0 2000 4001 8001 15998 23999 28000 32001 36001 40002 47999 60001 199997 (kilograms)

wtbins = 0 4410 8820 17640 35270 52910 61730 70550 79370 88190 105820 132280 440920 (pounds)

Notes	Except for *LowestBin*, all other bin boundaries mark the beginning of the **next** bin. A weight exactly equal to that boundary value will be recorded in the bin **below**. Vehicles whose weight falls outside the stated ranges will not be recorded.
Example	**wtbins = 0 5000 6000 7000 8000 9000 10000 20000**
Display syntax	**wtbins**
Example response	`WTBINS = 0 2000 5000 10000 20000 99999`
Keypad menu	`Start Survey / Bins, Weight` The display shows the current number of bins.
Related commands	INTSPEC is used to select this type of classification. Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtbins =** .....

Your notes

WTCAL

WeighT CALibration

Description	Each Weigh-in-Motion sensor has a calibration factor which relates the signal output to the measured weight. WTCAL sets these calibration factors or displays the ones currently in use.
Essential when	setting up a WIM layout.

Command syntax	**wtcal =** *Cal1* [*Cal2...*]
Parameters	*Cal1* is the calibration factor for the first sensor. *Cal2* etc. are calibration factors for any further sensors, in the order specified using SENSORS. The factors are obtained by calibration using WTCALIB. They are in arbitrary units which are not affected by the UNITS setting.
Valid range	1 to 9999, up to 16 values
Default	wtcal = 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500
Example *Explanation*	**wtcal = 521 489 398 643** WIM sensors are normally in groups of four, up to a maximum of 16
Display syntax	**wtcal**
Example response	`WTCAL = 525 520 409 555 500 500 500 500 500 500 500 500 500 500 500 500`
Explanation	WIM sensors are normally in groups of four, and WTCAL displays all 16 values.
Keypad menu	Not available

Related commands WTCALIB or WTCALVEH provide the calibration factors for vehicles of known axle weight. The user must average or adjust these values before re-entering them using WTCAL.

Auto-calibration using WTAUTCAL will overwrite any calibration factors entered manually using WTCAL.

Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtcal =** .....

Your notes

WTCALIB, WTCALIB+

WeighT CALIBration

Description Weigh-in-Motion sensors need to be calibrated using a vehicle of known wheel weights.

WTCALIB enters the wheel weights into the machine and can initiate a calibration test run. In the latter mode, WTCALIB produces an on-site printout of gross weight measurements and calibration factors. If time-tagged events are required as well, use WTCALIB+.

Full details of the use of WTCALIB are in the Golden River ***Weigh-in-Motion Highway Manual***. Only the bare syntax is given here for reference.

Essential when setting-up new weight sensors, and a single vehicle with measured axle weights is available. (Use WTCALVEH for a multi-vehicle calibration convoy.)

WTCALIB has three different syntax modes:

Load to enter the wheel weights

Display to show the weights as entered

Run to initiate a calibration run.

WTCALIB+ is only valid for **Run** mode.

Load syntax **wtcalib =** *Wheel1k Wheel1o Wheel2k Wheel2o*

[*Wheel3k Wheel3o...*]

Parameters *Wheel1k* and *Wheel1o* are the kerb-side and off-side wheel weights for axle 1. All weights are in the current weight UNITS, and pairs of wheel weights **must** be entered in the same sequence as the pair of half-width WIM sensors are wired into the machine. Normally this will be kerb-side (*k*) before off-side (*o*) as shown here.

Wheel2k and *Wheel2o* are the corresponding pair of wheel weights for axle 2.

For vehicles with more than two axles, continue entering pairs of wheel weights as required. The machine will recognise the total number of entries made (which must obviously be an even number) and will analyse the test run accordingly.

Valid range Any values

Example **wtcalib = 1560 1560 1580 1420**

Display syntax **wtcalib**

Example response

```
1555 1555 1578 1419
L  VEH    SPD GROSS   <------AXLE1------>     <------AXLE2------>
   NO        WEIGHT  WS1   WS2   WS3   WS4   WS1   WS2   WS3   WS4
01 001745 75  16281  428   335   336   315   366   248   330   221
```

Explanation The first line repeats the weight calibration factors previously entered (NB: there may be rounding errors due to internal units conversion).
In the table, the first column headed L is the lane number. The vehicle number, speed and gross weight are followed by the weights of axles 1 and 2, from weight sensors 1–4 (WS1–WS4).
WTCALIB+ would have produced a block of time-tagged event readings immediately below the table heading.

Run syntax **wtcalib** *Lane1* [*Lane2***...**]

Parameters *Lane1* is the first lane to be monitored during the calibration run. Entries for *Lane2* etc. are optional, up to a maximum of 8 lanes.

Notes As vehicles – including the test vehicle – pass by in the specified lane(s), the machine outputs an on-site printout of the weight details. If time-tagged events are required as well, use **wtcalib+** with the above syntax.

Having collected the on-site printout, it is up to the user to identify the readings from the test vehicle, analyse those results and calculate suitable calibration factors for entry using WTCAL.

Examples **wtcalib 1 4**

wtcalib+ 1 4

Keypad menu Not available

Related commands WTCAL is used to enter the calibration factors calculated from the results of a WTCALIB run.

Use WTCALVEH instead of WTCALIB if a multi-vehicle axle weight calibration convoy is available.

See the Golden River ***Weigh-in-Motion Highway Manual*** for full details of weight sensor calibration.

Your setting **wtcalib =** .....

Your notes

WTCALVEH

WeighT CALibration, multiple VEHicles

Description Each Weigh-in-Motion sensor must be calibrated using one or more vehicles with known axle weights.

WTCALVEH is an extended form of WTCALIB in its **load** mode (see page 256), which allows the input of axle weights for up to three vehicles. The user must average or adjust the results obtained, and then use WTCAL to enter a single calibration factor for each WIM sensor.

Essential when setting up a WIM layout, and a multi-vehicle axle weight calibration convoy is available.

Command syntax **wtcalveh** *Vehicle* **=** *Wheel1k Wheel1o Wheel2k Wheel2o...*

[*Wheel3k Wheel3o...*]

Parameters *Vehicle* is the number of the calibration vehicle (1 to 3).

Wheel1k and *Wheel1o* are the kerb-side and off-side wheel weights for axle 1. All weights are in the current weight UNITS, and pairs of wheel weights **must** be entered in the same sequence as the pair of half-width WIM sensors are wired into the machine. Normally this will be kerb-side (*k*) before off-side (*o*) as shown here.

Wheel2k and *Wheel2o* are the corresponding pair of wheel weights for axle 2. Use *Wheel3k* etc. if there are more than two axles.

Valid range *Vehicle* number: 1 to 3
Axle weights: any values

Defaults none

Example **wtcalveh 2 = 521 489 398 643**
Explanation WIM sensors are normally in groups of four, up to a maximum of 16

Display syntax **wtcalveh**

Example response

```
WTCALVEH 1 = 1147 1147 2093 2093
WTCALVEH 2 = 3740 3740 4030 4030 4763 4763
WTCALVEH 3 = 3230 3230 1940 1940 4390 4390 3110 3110 4060 4060
```

Keypad menu Not available

Related commands WTCALIB in its **load** mode is equivalent to WTCALVEH 1. WTCAL is used to enter the user-averaged calibration factors calculated from the results of a WTCALIB run.

Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your settings

wtcalveh 1 = ...

wtcalveh 2 = ...

wtcalveh 3 = ...

Your notes

WTCYCLES

WeighT scanning CYCLES

Description	Weight detection is carried out by scanning each weight sensor in succession. WTCYCLES determines how long each sensor is scanned before moving on to the next. Take great care when using this command: scanning too quickly or too slowly may cause vehicles not to be detected.
Essential when	using weight sensors, and when adjusting WTTHRES does not give sufficiently accurate results.

Command syntax	**wtcycles =** *Number*
Parameters	*Number* is in arbitrary units. High values give the quickest scanning rate, and low values the slowest.
Valid range	0 to 255
Default	wtcycles = 192
Example	**wtcycles = 190**
Display syntax	**wtcycles**
Example response	`WTCYCLES = 189`
Keypad menu	`Tech Mode / WTCYCLES`
Related commands	Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtcycles =** .....

Your notes

WTEVLEN

WeighT maximum EVent LENgth

Description If a vehicle remains static above a weight sensor, the detector will continue to give a signal each time it is scanned. Until that signal disappears, the vehicle will not be counted. To prevent this situation from 'jamming' the counter, it is advisable that the detection circuit resets itself after a certain period of time determined by WTEVLEN (or alternatively WTHOLD).

A suitable setting for WTEVLEN would be the largest number of sensor cycles for which you would expect a tyre to be over the sensor at normal speeds.

Essential when using weight sensors in locations where traffic is liable to be halted.

Command syntax **wtevlen =** *Cycles*

Parameters *Cycles* is the maximum allowable number of sensor cycles giving continuous detection, after which the weight sensor will reset itself. Sensor type automatically detected.

Suitable values would be 4 cycles for speeds above 50 km/hr, or 80 cycles for lower speeds.

Valid range 0 to 255

Default wtevlen = 40 [GRWIM] = 100 [Kistler WIM]

Example **wtevlen = 5**

Display syntax **wtevlen**

Example response `WTEVLEN = 60`

Keypad menu `Tech Mode / WTEVLEN`

Related commands WTEVLEN performs the same function as WTHOLD; changing either one also changes the other.

Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtevlen =** .....

Your notes

WTFILTER

WeighT FILTER

Description Weight sensors typically do not produce one pulse when the axle passes over, but several. To 'debounce' this signal, WTFILTER programs the detector card to define a 'dead time' after the first pulse, during which any further pulses generated by the same axle will be ignored.

Essential when Weight sensors are in use.

Command syntax **wtfilter =** *Time*

Parameters *Time* is the dead-time in milliseconds.

Valid range 1 to 100 milliseconds

Default wtfilter = 8

Example **wtfilter = 5**

Display syntax wt**filter**

Example response `WTFILTER = 7`

Keypad menu Not available

Related commands WTTHRES controls the WIM sensor threshold level.

Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtfilter =**

.....

Your notes

WTHOLD

WeighT detector HOLD time

Description	If a vehicle remains static above a weight sensor, the detector will continue to give a signal each time it is scanned. Until that signal disappears, the vehicle will not be counted. To prevent this situation from 'jamming' the counter, it is advisable that the detection circuit resets itself after a certain period of time determined by WTHOLD (or alternatively WTEVLEN). A suitable setting for WTHOLD would be the longest time you would expect a tyre to be over the 3cm wide sensor at normal speeds.
Essential when	using weight sensors in locations where traffic is liable to be halted.

Command syntax	**wthold =** *Time*
Parameters	*Time* is the maximum allowable period of continuous detection (in milliseconds), after which the weight sensor will reset itself. Suitable values would be 50 milliseconds for speeds above 50 km/hr, or 200 milliseconds for lower speeds.
Valid range	1 to 3000
Default	wthold = 50
Example	**wthold = 80**
Display syntax	**wthold**
Example response	`WTHOLD = 120`
Keypad menu	Not available
Related commands	WTHOLD performs the same function as WTEVLEN; changing either one also changes the other. Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wthold =** .....

Your notes

WTMAXAREA

WeighT MAXimum AREA

Description	As a wheel passes over a weight sensor, the signal produced is a broad, round-topped pulse. To distinguish this from false signals of similar peak height, the *Marksman 660* checks the total 'integrated area' of the signal. If this is less than WTMAXAREA, the signal is accepted. When using this command, take great care not to set the value too low, or else vehicles may not be detected.
Essential when	carrying out WIM surveys at an electrically noisy site.

Command syntax	**wtmaxarea =** *MaxArea*
Parameter	*MaxArea* is a value in arbitrary units, below which a signal will not be recognised as a WIM event.
Valid range	0 to 65535
Default	wtmaxarea = 50000
Example	**wtmaxarea = 55000**
Display syntax	**wtmaxarea**
Example response	`WTMAXAREA = 50000`
Keypad menu	`Tech Mode / WTMAXAREA`
Related commands	WTMINAREA; also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtmaxarea =** .....

Your notes

WTMINAREA

WeighT MINimum AREA

Description	As a wheel passes over a weight sensor, the signal produced is a broad, round-topped pulse. To distinguish this from narrower false signals of similar peak height, the machine checks the total 'integrated area' of the signal. If this is greater than WTMINAREA, the signal is accepted. When using this command, take great care not to set the value too high, or else vehicles may not be detected.
Essential when	carrying out WIM surveys at an electrically noisy site.

Command syntax	**wtminarea =** *MinArea*
Parameter	*MinArea* is a value in arbitrary units, below which a signal will not be recognised as a WIM event.
Valid range	0 to 65535
Default	wtminarea = 10
Example	**wtminarea = 12**
Display syntax	**wtminarea**
Example response	`WTMINAREA = 22`
Keypad menu	`Tech Mode / WTMINAREA`
Related commands	WTMAXAREA; also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtminarea =**
.....

Your notes

WTMODE

WeighT detector MODE

Description	WTMODE is not normally adjusted by the user; its main purpose is to change the computing load on the central processor for diagnostic purposes.
Essential when	using weight sensors with non-standard detection or processing systems, and for investigation of on-site problems.

Command syntax	**wtmode = 0 / 1**	
Parameters	**0**	Both ‘detect on’ and ‘detect off’ events will be passed to the main processor
	1	Only ‘detect off’ events will be reported (but with the time-tag of the ‘on’ event).
Valid range	0 or 1	
Default	wtmode = 1	
Example	**wtmode = 0**	
Display syntax	**wtmode**	
Example response	`WTMODE = 1`	
Keypad menu	`Tech Mode / WTMODE`	
Related commands	Also see the Golden River ***Weigh-in-Motion Highway Manual***.	

Your setting **wtmode =** .....

Your notes

WTOVERS

Description Weight sensors are subject to electrical noise which may produce false 'detect' signals of short duration. Each sensor is scanned repeatedly, and vehicle detection is only considered genuine after a number of successive scans have given a positive response.

WTOVERS sets or displays the number of successive positive responses required.

Essential when using weight sensors in electrically noisy environments.

Command syntax **wtovers =** *Number*

Parameters *Number* is the number of successive scans that must find a 'detect' condition in order for detection to be recorded.

Valid range 0 to 255

Default wtovers = 2

Example **wtovers = 8**

Display syntax **wtovers**

Example response `WTOVERS = 5`

Keypad menu `Tech Mode / WTOVERS`

Related commands Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtovers =** .....

Your notes

WTTHRES

WeighT sensor THREShold

Description	WTTHRES controls the detection threshold, the level of output from the weight sensor which is considered a valid detection signal.
Essential when	using weight sensors.

Command syntax	**wtthres =** *DetectLevel*
Parameters	*DetectLevel* is the detection threshold level (in arbitrary units). Sensor type automatically detected.
Valid range	0 to 255
Default	wtthres = 8 [GRWIM] = 50 [Kistler WIM]
Notes	The optimum threshold depends on the levels of electrical interference at the site.
Example	**wtthres = 10**
Display syntax *Example response*	**wtthres** `WTTHRES = 13`
Keypad menu	`Tech Mode / WTTHRES`
Related commands	WTFILTER sets the weight sensor debounce time in milliseconds. Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtthres =**

.....

Your notes

WTUNDERS

Description Weight sensors are subject to electrical noise which may include false ‘dropouts’ of short duration in the signal. Each sensor is scanned repeatedly and the vehicle is not considered to have gone past until a number of successive scans have produced no response.

WTUNDERS sets or displays the number of successive negative responses required to register ‘un-detection’.

Essential when using weight sensors in electrically noisy environments.

Command syntax **wtunders =** *Number*

Parameters *Number* is the number of successive scans that must find a ‘no response’ condition in order for ‘un-detection’ to be recorded.

Valid range 0 to 255

Default wtunders = 2

Example **wtunders = 8**

Display syntax **wtunders**

Example response `WTUNDERS = 5`

Keypad menu `Tech Mode / WTUNDERS`

Related commands Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtunders =** .....

Your notes

WTWINDOW

WeighT detection WINDOW

Description	Weight sensors are scanned repeatedly in a succession of short pulses. Because the WIM process is only initiated after signals exceed WTTHRES, some of the earlier, lower-level signal is missed. WTWINDOW specifies the number of scans to be added on to the beginning of the record to compensate for this effect.
Essential when	using weight sensors.

Command syntax	**wtwindow =** *Counts*
Parameters	*Counts* is the number of scan counts to be added on
Valid range	0 to 255
Default	wtwindow = 8
Example	**wtwindow = 18**
Display syntax	**wtwindow**
Example response	`WTWINDOW = 15`
Keypad menu	Tech Mode / WTWINDOW
Related commands	Also see the Golden River ***Weigh-in-Motion Highway Manual***.

Your setting **wtwindow =** .....

Your notes

Appendix 1 – GRPS Error and Warning Codes

Incorrect input will produce either an error message or a warning message.

If the *Marksman* has an LCD display, the message appears in the display window as a moving 'banner' (firmware version 1.98 onwards). Press any key to stop the error message.

The GRPS warning codes are given after the following list of error messages.

Entry errors

Error 01 **Unrecognised Command**
The first word entered was not recognised as a command; check that the command is correct.

Error 02 **Missing '='**
The '=' sign was expected because of the command format but found missing. Check the **Command syntax** and examples for this command.

Error 03 **Invalid Parameter**
The parameter given is not of the type required. Check the **Command syntax** and examples for this command.

Error 04 **Too many parameters**
The number of parameters was more than expected. Check the **Command syntax** and examples for this command.

Error 05 **Parameter out of range**
The parameter given lies outside the valid range. Check the upper and lower limits.

Error 06 **Parameter missing**
The number of parameters was fewer than expected. Check the parameter details for this command.

Error 07 **Illegal characters in parameter**
Found an unexpected character. Check the parameter details carefully.

Error 08 **Bins must be in ascending order**
Check that bin boundaries are all in ascending order.

File errors

Error 10 **No files specified**
A valid filename (see page 17) is required for this command.

Error 11 **No such file**
The filename given is not present in the machine. Check filenames using the DIR command.

Error 12 **Can't delete unretrieved or open file**
The command cannot be implemented because the file attribute is either 'U' (un-retrieved) or 'O' (open, in use).

Error 13 **File access error**
The reason for the file error cannot be determined. Check that you specified the correct filename and also that it is valid (see page 17). Possibly the file specified is of the wrong type or has become corrupted.

Memory errors

Error 014 **Memory full**

The available memory has become full, without the situation being anticipated by the MEMFULL setting.

Error 015 **Memory unformatted**
Memory in the *Marksman* machines is formatted like an MS-DOS RAMdisk. The error message appears if this formatting has been lost or has become corrupted (usually due to complete loss of main and backup battery power). A level 4 RESTART will restore the memory formatting.

Date and time errors

Error 20 **Invalid time**
Time must be in the range 00:00:00, to 23:59:59. Re-enter a correct value (seconds are optional).

Error 21 **Invalid date**
The date supplied is impossible, or not valid for the current setting of DATEFORM. Check input and also DATEFORM to confirm the format required. Year entries in two-digit format beyond AD2000 are interpreted correctly, and the year can also be entered in four-digit format.

Error 22 **Invalid date or time**
The machine cannot determine whether Error 20 or 21 occurred; there may have been no space separating the date and time.

Survey errors

Error 30 **Survey active**
The command you just gave is not allowed while a survey is in progress. If necessary, stop the survey, re-enter the command and re-start the survey.

Error 31 **Survey not active**
The command you just gave is not allowed unless a survey is in progress. The input prompt must not be Q> (quiescent machine).

Error 32 **No sensor configuration active**
The command you just gave cannot be used unless a sensor configuration is active; use the SENSORS command to specify the sensor array.

Error 33 **Please specify survey type**
The STARTREC command requires the parameter **int**, **vbv**, or **both.** Check the **Command syntax** entry for STARTREC.

Error 39 **EUR6 can only be used with LL configuration**
See information on CLASS.

Communications errors

Error 50 **Invalid baud rate**
The baud rate specified is not available. Check details on COM1 command for permissible values.

Error 51 **Invalid number of bits**
The number of bits specified is not available; only 7 or 8 bits are permissible. Check details on COM1 command.

Error 52 **Invalid parity**
The parity specified is not available in combination with the specified number of bits. Check details on COM1 command.

Error 53 **Invalid flow control**
The flow control method specified is not available. Check details on COM1 command.

Other errors

Error 60 **Parameter incompatible with fitted sensor cards**
The SENSORS and/or INTSPEC settings are not compatible with the sensor cards fitted to the machine (see tables on pages 186 and 113).

Error 61 **Please specify lane number or GAS**
The MONITOR command requires one of these two parameters.

Error 62 **Please specify GENERAL or SURVEY**
The SETUP command requires one of these two parameters.

Error 63 **Peak interval is not divisible by the interval**
The peak interval time (PEAKINT command) must be exactly divisible by the INTERVAL time. Check details on both commands.

Error 64 **Interval is not divisible by the break period**
The time specified by INTERVAL must be an exact sub-multiple of the period between file breaks (set using BREAK); or it can be the same as the BREAK period. Check details on both commands.

Error 65 **Value is not divisible by 24 hours**
The interval and peak interval times must each be an exact sub-multiple of 24 hours. Check details on INTERVAL and PEAKINT commands.

Error 70 **Incorrect password**
The incorrect password has been used when changing to either **Read-only** mode or **Read-write** mode. Check details on the INTERFACE and PASSWORD commands.

Error 71 **Command not accessible in this mode**
The command just attempted is not available because the machine is in either **Read-only** or **Locked** mode. Check details on the INTERFACE command.

Error 73 **File not accessible in this mode**
You have tried to use the PRINT command, which is not allowed when the machine is in **Read-only** mode. Check details on the INTERFACE command.

Error 99 **Command unavailable on this machine**
The command just attempted is a valid GRPS command, but is not available on this model of machine.

Warning messages

Warning 01 **String has been truncated**
The text just entered was too long, and has been truncated. Enter the same command again without any parameters, to display the truncated text that has been stored.

Warning 02 **Can only retrieve 10 files at a time**
A multiple-file RETRIEVE option has selected more than 10 files. After the first 10 files have been retrieved, you will need to use RETRIEVE again to download the others you require.

Appendix 2 – GRPS ASCII Print Formats

The following examples show the output formats produced by the PRINT and OSP commands.

The corresponding OSP+ formats include an additional header line, and include time-tagged events between vehicle printouts.

The first group are the three available Interval file formats; the second group are the four available Vehicle by Vehicle (VBV) formats.

Text in this Appendix is compressed to show the true layout of columns. On narrow displays or printouts, long lines may be wrapped-around, making the column arrangement less clear than shown here.

Structure

All output formats start with a block of information lines, each beginning with a `*`. The first line is always `* BEGIN`, and the last line of the whole file is always `* END` followed by a hexadecimal checksum which is used to conform the integrity of the file.

The `* FORMAT` line states which format is being used.

The `* FORMATTER` line shows how the data were converted from the internal binary format to the ASCII printable format. If the data came directly from the *Marksman* using the PRINT command, the `* FORMATTER` line gives the type of *Marksman* (`660`, `360`, `400` etc.) and the firmware version used. Other possible sources of ASCII or printed data include the GRFORMAT and GRFORM1 utility programs, and the *Showman* range of data analysis programs.

Most of the other information lines record the type of `INSTRUMENT` and the GRPS command settings that were used to obtain the data.

The line beginning `* HEAD` shows the headings for the columns of numerical data that will follow. (`HEAD` also appears elsewhere as an abbreviation for ‘headway’.)

Abbreviations

Abbreviations are used in printout formats as space is limited. Alternatives for some formats are shown in [ ].

AW1 to AW9	Axle weights (axles 1 to 9)
AX	Number of axles
C	Channel
CN	Vehicle count
CS [CL]	Vehicle class (Dependant on class scheme selected)
D [DN] [DR]	Vehicle Direction
DDMMYY	Day/month/year
GAP [GP]	Inter-vehicle Gap
GS	CO1 gas level
GWTOT [GW]	Vehicle Gross weight
* HEAD	Start of the Heading list (in VBV files this first column is the vehicle number)
HEAD [HD][HW]	Headway between vehicles
HH	Hundredths of a second *
HHMM	Hours:minutes
L [LA]	Lane number
LENTH [LN]	Vehicle Length
RESCOD	Vehicle result code *
SPD [SP]	Vehicle Speed
SS	Seconds
W1-2	Axle separation (axle 1 to 2)
W2-3	Axle separation (axle 2 to 3)
W3-4	Axle separation (axle 3 to 4)
W4-5	Axle separation (axle 4 to 5)
W5-6	Axle separation (axle 5 to 6)
W6-7	Axle separation (axle 6 to 7)
W7-8	Axle separation (axle 7 to 8)
W8-9	Axle separation (axle 8 to 9)
WBTOT [WB]	Wheelbase

* The HH (hundredths of a second) and RESCOD (results code) fields are only displayed by the OSP and OSP+ commands. To save space they are not stored in a VBV file, and the PRINT command and GRFORMAT will show zeros in these columns.

TTE output

Time-tagged events (TTEs) generated by detector cards are displayed in OSP+ printout, and can be used for diagnostic purposes. TTEs are also stored in VBV files if TTEFILTER has been changed from its default setting of none to a valid condition for TTE recording.

Details of the additional output for TTEs are given on page 300, and include the additional TTE formatting for the *Marksman 410*.

Overruns

Data fields in all the printout formats are large enough to accommodate all practical values. In the unlikely event that a measured value is too large to fit the data field, the printout will show ‘all 9s’ within the space available.

Interval format 1

Example command: print 1 example.i00

```
* BEGIN
* FORMAT = INT-1
* FORMATTER = 660 Release = 1.95
* INSTRUMENT = 660 Serial = 1234567 Release = 1.95
* FILENAME =
* SITE =
* LOCATION =
* GRIDREF =
* HEADINGS =
* STARTREC = 15:21 15/01/96
* STOPREC = 22:45 15/01/96
* BATTERY = 6.40 6.40
* SENSORS = TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2
* DATEFORM = DD/MM/YY
* UNITS = Metric
* INTERVAL = 15
* PEAKTIME = 00:00 00:00 00:00 00:00 00:00 00:00
* PEAKINT = 5
* CHANNELS = 1 1 1 1 1 1 1 1
* INTSPEC = CNT
* INTFILTER = ALL
* HEAD HHMM C ...0 ...1 ...2 ...3 ...4 ...5 ...6 ...7 ...8 ...9 ..10 ..11
* INTERVAL = 15
150196 1530 1 0003 0007 0011 0015 0019 0023 0027 0031 0035 0039 0043 0047
150196 1830 1 0051 0055 0059
* END 26 FFFF
```

Interval format 2

Example command: print 2 example.i00

```
* BEGIN
* FORMAT = INT-2
* FORMATTER = 660 Release = 1.95
* INSTRUMENT = 660 Serial = 1234567 Release = 1.95
* FILENAME =
* SITE =
* LOCATION =
* GRIDREF =
* HEADINGS =
* STARTREC = 15:21 15/01/96
* STOPREC = 22:45 15/01/96
* BATTERY = 6.40 6.40
* SENSORS = TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2
* DATEFORM = DD/MM/YY
* UNITS = Metric
* PRUNITS = KPH-CM-10KG
* INTERVAL = 15
* PEAKTIME = 00:00 00:00 00:00 00:00 00:00 00:00
* PEAKINT = 5
* CHANNELS = 1 1 1 1 1 1 1 1
* INTSPEC = CNT
* INTFILTER = ALL
* HEAD HHMM C       CN 1
* INTERVAL = 15
150196 1530 1 00 00 0001
150196 1545 1 00 00 0002
150196 1600 1 00 00 0003
150196 1615 1 00 00 0004
150196 1630 1 00 00 0005
```

(similar lines omitted)

```
150196 2145 1 00 00 0026
150196 2200 1 00 00 0027
150196 2215 1 00 00 0028
150196 2230 1 00 00 0029
150196 2245 1 00 00 0030
* END 55 FFFF
```

Interval format 3

Example command: print 3 example.i00

```
20,"BEGIN"
22,"FORMAT","INT-3"
23,"FORMATTER","660","Release",1.95
24,"INSTRUMENT","660","Serial",1234567,"Release",1.95
25,"FILENAME",""
26,"SITE",""
27,"LOCATION",""
31,"GRIDREF",""
32,"HEADINGS","","","","","","","","",""
72,"STARTREC",15,21,15,01,96
73,"STOPREC",22,45,15,01,96
28,"BATTERY",6.40,6.40
29,"SENSORS","TT*2","TT*2","TT*2","TT*2","TT*2","TT*2","TT*2","TT*2"
30,"DATEFORM","DD/MM/YY"
89,"UNITS","Metric"
80,"PRUNITS","KPH-CM-10KG"
40,"INTERVAL",15
43,"PEAKTIME",00,00,00,00,00,00,00,00,00,00,00,00
43,"PEAKINT",5
41,"CHANNELS",1,1,1,1,1,1,1,1
42,"INTSPEC","CNT"
64,"INTFILTER","ALL"
39,"HEAD HHMM C       CN 1"
40,"INTERVAL",15
150196,1530,1,00,00,0001
150196,1545,1,00,00,0002
150196,1600,1,00,00,0003
150196,1615,1,00,00,0004
150196,1630,1,00,00,0005
```

(similar lines omitted)

```
150196,2145,1,00,00,0026
150196,2200,1,00,00,0027
150196,2215,1,00,00,0028
150196,2230,1,00,00,0029
150196,2245,1,00,00,0030
21,"END",55,"FFFF"
```

Vehicle by Vehicle format 1

Example command: print 1 example.v00

```
* BEGIN
* FORMAT = VBV-1
* FORMATTER = 660 Release = 1.95
* INSTRUMENT = 660 Serial = 1234567 Release = 1.95
* FILENAME =
* SITE =
* LOCATION =
* GRIDREF =
* HEADINGS =
* STARTREC = 15:24 15/01/96
* STOPREC = 15:24 15/01/96
* BATTERY = 6.40 6.40
* SENSORS = TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2
* DATEFORM = DD/MM/YY
* UNITS = Metric
* PRUNITS = KPH-CM-10KG
* CLASS = EUR13
* VBVFILTER = ALL
* HEAD DDMMYY HHMM SS HH RESCOD  L D HEAD  GAP SPD LENTH AX CS WBTOT W1-2 W2-3 W
3-4 W4-5 W5-6 W6-7 W7-8 W8-9
000001 150196 1524 23 00 000000  5 1 99.9 99.9  76  250  2  1   200  200
000002 150196 1524 23 00 000000  1 1 99.9 99.9  85  163  2 13   113  113
000003 150196 1524 24 00 000000  5 2  1.0  0.8  29  825  5 13   775  194  243
142  196
000004 150196 1524 24 00 000000  3 2 99.9 99.9  41  262  2  1   212  212
000005 150196 1524 25 00 000000  5 1 99.9 99.9  82  292  2  1   242  242
000006 150196 1524 25 00 000000  3 1  1.0  0.7  84  296  2  1   246  246
000007 150196 1524 26 00 000000  4 2 99.9 99.9  20  283  2  1   233  233
000008 150196 1524 26 00 000000  4 1 99.9 99.9  35  256  2  1   206  206
000009 150196 1524 28 00 000000  4 2  2.0  1.7  22  274  2  1   224  224
000010 150196 1524 30 00 000000  6 2 99.9 99.9  90  295  2  1   245  245
* END 30 FFFF
```

Vehicle by Vehicle format 2

Example command: print 2 example.v00

```
20,"BEGIN"
22,"FORMAT","VBV-2"
23,"FORMATTER","660","Release",1.95
24,"INSTRUMENT","660","Serial",1234567,"Release",1.95
25,"FILENAME",""
26,"SITE",""
27,"LOCATION",""
31,"GRIDREF",""
32,"HEADINGS","","","
```

Vehicle by Vehicle format 3

Example command: print 3 example.v00

```
* BEGIN
* FORMAT = VBV-3
* FORMATTER = 660 Release = 1.95
* INSTRUMENT = 660 Serial = 1234567 Release = 1.95
* FILENAME =
* SITE =
* LOCATION =
* GRIDREF =
* HEADINGS =
* STARTREC = 15:24 15/01/96
* STOPREC = 15:24 15/01/96
* BATTERY = 6.40 6.40
* SENSORS = TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2
* DATEFORM = DD/MM/YY
* UNITS = Metric
* PRUNITS = KPH-CM-10KG
* CLASS = EUR13
* VBVFILTER = ALL
* HEAD ------------------------------------------------
000001 150196 15:24:23.00 000000 LA=5 DN=1 SP=76 GP=409.5 HD=409.5
AX=2  CS=1  MTR  2.00   2.00
OK       TNE  2.15   1.21  0.93
000004 150196 15:24:23.00 000000 LA=1 DN=1 SP=85 GP=409.5 HD=409.5
AX=2  CS=13 MTR  1.13   1.13
OK       TNE  1.15   0.65  0.50
000007 150196 15:24:24.00 000000 LA=5 DN=2 SP=29 GP=0.8 HD=1.0
AX=5  CS=13 MTR  7.75   1.94  2.43  1.42  1.96
OK       TNE 21.36   1.67  1.76  7.54  4.60  5.79
000010 150196 15:24:24.00 000000 LA=3 DN=2 SP=41 GP=409.5 HD=409.5
AX=2  CS=1  MTR  2.12   2.12
OK       TNE  2.85   1.51  1.33

Similar lines deleted

000022 150196 15:24:26.00 000000 LA=4 DN=1 SP=35 GP=409.5 HD=409.5
AX=2  CS=1  MTR  2.06   2.06
OK       TNE  2.10   0.97  1.13
000025 150196 15:24:28.00 000000 LA=4 DN=2 SP=22 GP=1.7 HD=2.0
AX=2  CS=1  MTR  2.24   2.24
OK       TNE  3.30   1.59  1.71
000028 150196 15:24:30.00 000000 LA=6 DN=2 SP=90 GP=409.5 HD=409.5
AX=2  CS=1  MTR  2.45   2.45
OK       TNE  2.79   1.34  1.45
* END 50 FFFF
```

Vehicle by Vehicle format 4

Example command: print 4 example.v00

```
* BEGIN
* FORMAT = VBV-4
* FORMATTER = 660 Release = 1.95
* INSTRUMENT = 660 Serial = 1234567 Release = 1.95
* FILENAME =
* SITE =
* LOCATION =
* GRIDREF =
* HEADINGS =
* STARTREC = 15:24 15/01/96
* STOPREC = 15:24 15/01/96
* BATTERY = 6.40 6.40
* SENSORS = TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2 TT*2
* DATEFORM = DD/MM/YY
* UNITS = Metric
* PRUNITS = KPH-CM-10KG
* CLASS = EUR13
* VBVFILTER = ALL
* HEAD   L GP SPD AX CL LEGAL   MTRS.3...5...7....10........15........20
15:24:23 5 99  76  2  1 OK      1--1
15:24:23 1 99  85  2 13 OK      h-h
15:24:24 5  1  29  5 13 OK      2--2---8-5--6
15:24:24 3 99  41  2  1 OK      2---1
15:24:25 5 99  82  2  1 OK      2---1
15:24:25 3  1  84  2  1 OK      1---1
15:24:26 4 99  20  2  1 OK      1---1
15:24:26 4 99  35  2  1 OK      1---1
15:24:28 4  2  22  2  1 OK      2---2
15:24:30 6 99  90  2  1 OK      1---1
* END 30 FFFF
```

TTE (Time Tagged Event) Format

Time-tagged events (TTEs) generated by detector cards are displayed in OSP+ printout, and can be used for diagnostic purposes.

TTEs are also stored in VBV files if TTEFILTER has been changed from its default setting of none to a valid condition for TTE recording. See the TTEFILTER command description for further details.

When a VBV file with TTE data is displayed by the PRINT command or using the GRFORMAT program, the data format is as shown below. This format is the same as used by the OSP+ command to display TTE data.

The format of TTE header line and data line is as follows:

```
* HEAD T E TIME-TAG AREA  NORMAL CURENT PEAK
0 1 xx s e ttttttt aaaaa nnnnnn cccccc pppppp
```

Explanation:

`0 1` These are always the first two characters

`xx` Sensor number (1-99)

`s` Sensor type
- `0` = Tube
- `1` = Loop
- `2` = WIM strip
- `3` = Axle strip
- `4` = Piezo
- `5` = Video
- `6` = Pollution monitor
- `7` = Bicycle detecting tube

`e` Event type
- `0` = Sensor activated
- `1` = Sensor deactivated
- `2` = Sensor trace
- `9` = Pollution update

`ttttttt` Event time-tag

In the columns headed AREA, NORMAL, CURENT and PEAK, numerical data are only displayed if applicable to the type of sensor involved:

AREA	aaaa	Area under curve at this point
NORMAL	nnnnnn	Sensor reference level
CURENT	cccccc	Sensor current count level
PEAK	pppppp	Peak count level

Marksman 410 OSP+ format

The Tyre Contact Width (TCW) tube sensors in the *Marksman 410* have a special OSP+ format. The sensor type number, displayed in the fourth field of all event lines, number is 7 for TCW sensors (standard tube sensors are type 0).

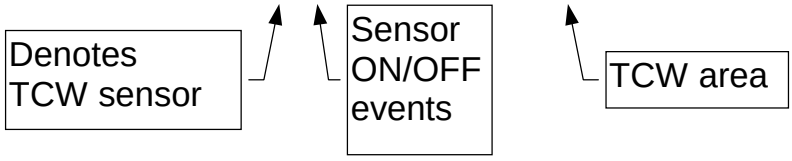
TCW tube sensors generate lines of output in pairs, one for the *Sensor On* event and a second line for the separate *Sensor Off* event (standard tube sensors only generate *Sensor On* events). The event number is displayed in the fifth field of each event line. *Sensor On* events are numbered 0, and *Sensor Off* events are numbered 1.

The tyre contact width is stored in the area field of the *Axle Off* event. The area field is displayed in the seventh field of each *Axle Off* event line.

Output

These are the two event lines generated by OSP+ from the *Marksman 410* for a single tyre crossing:

```
0 1 01 7 0 08167424 00000 000000 000000 000000
0 1 01 7 1 08167801 00089 000000 000000 000000
```



Appendix 3 – ASCII Codes

These codes are used with the EOFCHARS, EOLCHARS and EOPCHARS commands.

Symbol or Character
Null
SOH (Start of Heading)
STX
ETX
EOT
ENQ (Enquiry)
ACK
BEL (Bell)
BS (Back Space)
LF (Line-feed)
FF (Form Feed)
CR (Carriage Return)
DC1 (Device Control 1)
DC2 (Device Control 2)
DC3 (Device Control 3)
DC4 (Device Control 4)
ESC (Escape)
Numerals 0..9
Letters A..Z
Letters a..z
DEL (Delete)

Appendix 4 – Data Module Commands

Overview

The Golden River *Data Module* is a compact, portable memory unit which retrieves traffic data from the *Marksman 660* or other compatible Golden River instruments, and can also be configured used to issue a predefined set of commands to every unit to which it is connected.

The *Data Module* is designed to connect either to a *Marksman* unit or to a PC running terminal emulation software.

When connected to a *Marksman*, the *Data Module* can issue GRPS commands to retrieve files into its own memory; this can either take place automatically, retrieving all Un-retrieved files, or selected files can be retrieved using a file called COMMAND.LST.

When connected to a PC terminal, the *Data Module* responds to the DIR and RETRIEVE commands to list and download the stored files. These will include a log file of the *Data Module*'s activities (see below). The UPLOAD command is used to transfer files from the PC terminal to the *Data Module*, particularly the command file COMMAND.LST which has a special format.

Data Module Log File

After each connection to either a *Marksman* unit or a PC terminal, the *Data Module* updates its log file of activities. This file is called DM_LOG.*nnn*, where *nnn* is an automatically-generated number incrementing from 001 to 999.

The DM_LOG file contains information about the units to which the *Data Module* has been connected, the commands issued and responses received, and the file transfers that have taken place or were attempted.

Data Module Command Set

The *Data Module* recognises the following GRPS commands, using the standard syntax as described in the main text of this manual.

ALL	Displays a list of the commands available

DEL	Deletes selected files
DIR	Lists the files in the *Data Module*'s memory
PRINT	Prints selected files, e.g. DM_LOG and COMMAND.LST
RESTART	Restarts the *Data Module* without erasing any stored files
RETRIEVE	Downloads files from the *Data Module* to the PC terminal, using Ymodem batch protocol.
UPLOAD	Uploads files from the PC terminal to the *Data Module*, using Ymodem batch protocol.

Further Data Module commands

The following commands are only usable with the *Data Module*; they are not recognised by the *Marksman* or other GRPS units.

EXECUTE ...TO	Deliver a series of GRPS commands from a named ASCII text file in the *Data Module* to a machine at a specified SITE or LOCATION, or to all machines to which the *Data Module* is connected.
RETRIEVE ...FROM	For use only in COMMAND.LST: instructs the *Data Module* to retrieve a named file from a machine at a specific SITE or LOCATION.
SEND ...TO	For use only in COMMAND.LST: instructs the *Data Module* to send a named file to a machine at a specific SITE or LOCATION.

These three commands are described below, in the standard GRPS format of this manual.

Important notes

RETRIEVE...FROM is totally different from the normal GRPS command RETRIEVE. Likewise, SEND...TO is totally different from UPLOAD.

Marksman units will respond to RETRIEVE...FROM, SEND...TO or EXECUTE...TO with the error message:

```
Error 01 Unrecognised Command.
```

The *Data Module* cannot handle RETRIEVE...FROM or SEND...TO as direct commands, but only within a COMMAND.LST file.

EXECUTE ...TO

Command syntax **execute** [*Filename*] **to site** = [*Site*]

execute [*Filename*] **to location** = [*Location*]

execute [*Filename*] **to all**

Parameters *Filename* is a valid MS-DOS/GRPS filename (up to 8 characters). The file must contain a list of valid GRPS commands (one per line).

Site and *Location* must be **exactly** as the relevant *Marksman* unit will respond to the SITE or LOCATION command (except that text format is not case-sensitive).

Note Remember that *Location* can be any text string, but *Site* may not include space characters.

Example 1 **execute siteone.ins to site = 1**
Explanation Sequentially executes the instructions in SITEONE.INS, but only to a machine that responds to the *Data Module*'s enquiry with `SITE = 1`.

Example 2 **execute instruct.lst to all**
Explanation Sequentially executes the instructions in INSTRUCT.LST each time the *Data Module* is connected to a *Marksman* unit.

RETRIEVE...FROM

SEND...TO

Command syntax **retrieve** [*Filename*] / **all** / **new** **from** **site** = [*Site*]

retrieve [*Filename*] / **all** / **new** **from** **location** = [*Location*]

send [*Filename*] **to** **site** = [*Site*]

send [*Filename*] **to** **location** = [*Location*]

Parameters

Filename is a valid MS-DOS/GRPS filename (up to 8 characters).

all will retrieve all files from the machine's memory, if possible.

new will retrieve all previously unretrieved files from the machine's memory, if possible.

Site and *Location* must be **exactly** as the relevant *Marksman* unit will respond to the SITE or LOCATION command (except that text format is not case-sensitive).

Note

Remember that *Location* can be any text string, but *Site* may not include space characters.

Example 1

retrieve new from site = 1234-5678

Example 2

retrieve all from location = high street 2

Example 3

send m660.cfg to site = 33003300

Appendix 5 – Release Notes since Issue 2.0

Issue 2.1

This issue corresponds to *Marksman 660* Firmware version 1.95.

☐ **New commands added**

CLOCK, TEST SYSAUDIT and TEST LPSTATUS.

Keypad tables and print formats updated.

Valid range for EOVDSEP increased to 2000 cm.

Default value of WTWINDOW changed to 8.

INTSPEC=WBT now allowed.

LPCYCLES, LPOVERS, LPTHRES and LPUNDERS can all be given different settings if two loop boards are fitted in a *Marksman 660*.

Reference to long input lines and LAN as an INTSPEC parameter removed.

Minor editorial changes.

Issue 2.2

This issue covers all *Marksman* GRPS commands, including those for the *Marksman 360*, *Marksman 400* and *Marksman 410.*

This issue corresponds to *Marksman* firmware version 1.98.

☐ **New commands added**

Chassis Height: CHAUTAVLOW (=LPCLS2), CHAUTCNT (=CLS2CNT), CHCALC, CHFIELD, CHHIGH% (=LP2THRES) and CHUSRAVLOW.

Weight: WTCALVEH, WTMAXAREA.

Marksman 410: TCWCALIB, TCWNOISE, TCWTHRES; enhancements to CLASS, MONITOR and OSP+.

☐ **Commands modified**

CLASS, INTSPEC, PRINT, PRUNITS, SENSORS, TEMPCAL, WTCAL, WTEVLEN, WTMINAREA.

☐ **Other enhancements**

- Four new sensor configurations – see SENSORS.
- 83kHz loop card supported.
- Keypad interface includes `Status / Last Vehicle`, to monitor the last vehicle in either direction.
- Vehicle monitor menus display vehicle gap.
- Full error messages displayed as continuously scrolling ‘banners’.
- Improved loop counting.
- More convenient valid ranges for some parameters.
- Support for Kistler WIM sensors included.

Other minor language and firmware updates.

Minor editorial changes in manual.